

CRWR Online Report 97 #7

GIS Based Reservoir Planning for the Souss Basin, Morocco

by

Kwabena Asante, M.S.

Graduate Research Assistant

and

David Maidment, PhD.

Principal Investigator

December 1997

CENTER FOR RESEARCH IN WATER RESOURCES

Bureau of Engineering Research • The University of Texas at Austin
J.J. Pickle Research Campus • Austin, TX 78712-4497

This document is available online via World Wide Web at
<http://www.ce.utexas.edu/centers/crwr/reports/online.html>

GIS Based Reservoir Planning for the Souss Basin, Morocco

APPROVED BY
SUPERVISING COMMITTEE:

David R. Maidment

Charles McGinnis

G. Edward Gibson, Jr.

Acknowledgements

I wish to thank Dr. David R. Maidment, Gen. Charles McGinnis and Dr. G. Edward Gibson who supervised and guided me through this thesis research. The time and other resources that you invested in me is sincerely appreciated. I would also like to thank the members of the GISHYDRO research team, in particular, Dr. David R. Maidment, Zichuan Ye and Seann Reed who were always ready to offer assistance when I needed it. I would also like to thank the staff of the DGH in Morocco, FAO and other agencies who provided the data for this study. Finally, I would like to thank my parents who made it all possible. Thank you for your support.

Kwabena Asante

August, 1997

GIS Based Reservoir Planning for the Souss Basin, Morocco

by

Kwabena Oduro Asante, M.S.E.

The University of Texas at Austin, 1997

David R. Maidment

This research was undertaken to develop a map-based surface water simulation model of a river basin, incorporating the operation of reservoirs. The Souss River in Morocco and the three major reservoirs located within the basin were used as the study case in the research which focused on simulating the effects of reservoirs on flow in the basin. A surface water simulation program developed at the University of Texas at Austin was customized to enable it to simulate the operation of reservoirs with variable monthly demands. The program was developed in the object-oriented programming language, Avenue, of the GIS software ArcView. Rainfall data from 48 rain gages stations spanning a 60 year period from 1935 to 1995 were processed into daily and monthly time series. Flow time series were also generated from runoff data collected at 6 gauging stations in the basin. Historical operation data from each of the three Souss reservoirs were also processed into time series and incorporated into the simulation to enable basin response to be predicted accurately. The effects of temporal scale on modeling effort and results were also explored by performing daily and monthly simulations. An ArcView extension called DAMS was constructed to enable the user to insert and remove dams as required. The extension includes preprocessing programs for creating reservoir operation and regulation tables, varying monthly water allocations and setting initial reservoir storage. It also contains a postprocessor for

generating a summary of performance of the reservoirs during the simulation period. The reservoir simulation was successfully completed and recorded storage levels were successfully reproduced using inflow data recorded in the reservoir operation tables. However, reservoir inflows generated by the surface water simulation model did not closely resemble those recorded in the operation tables, particularly during periods of highly variable flow.

Table of Contents

Chapter 1: Introduction.....	1
1.1 Project Background.....	1
1.2 Motivation.....	6
1.3 Objectives.....	8
1.4 Methodology.....	10
1.5 Contribution to Knowledge.....	12
Chapter 2: Literature Review.....	13
2.1 Introduction.....	13
2.2 Surface Water Simulation.....	13
2.3 Reservoir Simulation.....	14
2.4 GIS as a Planning Tool.....	17
Chapter 3: Data Processing for the Souss Basin.....	19
3.1 Preparing the Data.....	19
3.2 Developing River Basin Coverages.....	23
3.3 Processing Rainfall Data.....	47
3.4 Processing Runoff Data.....	54
3.5 Preparing Reservoir Data.....	61
Chapter 4: Modeling the Souss Basin.....	77
4.1 Developing the Model.....	78
4.2 Calibrating Model Parameters.....	92
4.3 Constructing Reservoir Simulation.....	107
4.4 Checking Model Reliability.....	113
4.5 Using the Model as a Planning Tool.....	117
4.6 Developing the DAMS extension.....	121
Chapter 5: Simulation Results.....	134
5.1 Parameter Calibration Results.....	134
5.2 Observed versus Simulated Flows.....	136
5.3 Reservoir Simulation Results.....	140

Chapter 6: Discussion and Conclusions.....	146
6.1 Discussion	146
6.2 Conclusions.....	152
Bibliography.....	154
Appendix 1: Data Dictionary.....	158
Appendix 2: Converting daily to monthly rainfall.....	162
Appendix 3: Converting daily to monthly runoff.....	168
Appendix 4: Interpolating for Missing Rainfall Records, Cmprain.pre.....	175
Appendix 5: Computing Surplus Runoff, Cmpsurp.pre.....	180
Appendix 6: Script for Inserting Dams, Insertdm.ave.....	188
Appendix 7: Script for Removing Dams, Removedm.ave.....	189
Appendix 8: Script for Making Water Allocations, Setdmval.ave.....	197
Appendix 9: Script for Setting Up Regulation Tables, Setdmrg.ave.....	204
Appendix 10: Script for Initializing Reservoir Storage, Setdmst0.ave.....	211
Appendix 11: Script For Summing Up Monthly Outflows, Cmptotal.ave....	214
Appendix 12: Script for Computing Simulation, Cmpsumry.ave.....	231
Appendix 13: Reservoir Simulation Script, Damrt.ave.....	240
Vita.....	253

List of Tables

Table 3.1: Sosprj Projection File.....	28
Table 3.2: Cell Values in the Flow Direction Grid	30
Table 3.3: The Hydrost.dat File of Runoff Station Coordinates	42
Table 3.4: Stattrib.txt file For Attributing Runoff Stations.....	43
Table 3.5: Attributes of Souss Basin Point Coverages	46
Table 3.6: Sample Daily Rainfall Data File.....	52
Table 3.7: Sample Monthly Rainfall Data File.....	53
Table 3.8: Souss Basin Runoff Data Files.....	54
Table 3.9: One Dimensional Runoff Data File.....	57
Table 3.10: Characteristics of the Souss Basin Reservoirs.....	62
Table 3.11: Reservoir Regulation File for Aoulouz.....	65
Table 3.12: Reservoir Regulation File for Abdel Moumen.....	66
Table 3.13: Reservoir Regulation File for Dkhila.....	67
Table 3.14: Translation of Reservoir Operation Fields.....	69
Table 3.15: Uncontrolled Monthly Releases at Abdel Moumen.....	70
Table 3.16: Original and Resampled Reservoir Operation Fields.....	72
Table 3.17: Parameter Description for Dam Operation Table.....	74
Table 3.18: Reservoir Operation Table for Abdel Moumen.....	75
Table 4.1: Request and Inputs for the Model Preprocessing.....	83
Table 4.2: Requests and Inputs for the Surplus Interpolation.....	87
Table 4.3: Surface Water Simulation Parameters.....	91
Table 4.4: Requests and Inputs for Setting Optimization Target.....	99
Table 4.5: Flow Routing Parameters.....	102
Table 4.6: Requests and Inputs for Setting Up Optimization Model.....	105
Table 4.7: Suggested Default Values for Routing Parameters.....	106
Table 4.8: Dams.dbf Table of Reservoir Attributes.....	108
Table 4.9: Reservoir Simulation Fields in Dam(id).dbf Tables.....	112
Table 4.10: Setting Up Dam(id).dbf Tables.....	124
Table 4.11: Source and Target Fields for Setting Up Dam(id).dbf.....	131
Table 4.12: Reservoir Simulation Summary Fields in Summary.dbf.....	133
Table 5.1: Comparison of Daily and Monthly Parameters.....	134
Table 5.2: Comparison of Souss and Niger Basins Parameters.....	136

List of Figures

Fig. 1.1: Map of the Major Rivers and Lakes of Africa.....	4
Fig. 1.2: Developing a Map based Surface Water Simulation Model.....	11
Fig. 3.1: Souss Database Structure.....	20
Fig. 3.2: Delineated Streams and Watersheds.....	34
Fig. 3.3: A Typical Three dimensional data structure.....	48
Fig. 3.4: Souss Basin Rain Gauging Stations.....	50
Fig. 3.5: Souss Basin Runoff Stations.....	55
Fig. 3.6: Graph of the Runoff Stations Period of Record.....	56
Fig. 3.7: Souss Basin Dams and Aquifer	63
Fig. 4.1: Flow Diagram of Model Preprocessing.....	79
Fig. 4.2: Diagram of Surface Water Menu.....	82
Fig. 4.3: Interpolation to Centroid of Watershed Polygons.....	88
Fig. 4.4: Chart of Daily Runoff at Amsoul.....	93
Fig. 4.5: Chart of Daily Rainfall at a Station near Amsoul.....	94
Fig. 4.6: Chart of Monthly Rainfall and Runoff at Aoulouz.....	95
Fig. 4.7: Monthly Runoff at 5 Souss Basin Runoff Stations in 1987.....	96
Fig. 4.8: Monthly Rainfall at Selected Souss Basin Stations in 1987.....	97
Fig. 4.9: Calibrating Flows at Aoulouz.....	104
Fig. 4.10: Relation of Dam Data Files	107
Fig. 4.11: Flow Diagram of Modified Preprocessing Procedure.....	114
Fig. 4.12: Inserting Dams from the DAMS menu.....	123
Fig. 4.13: Removing Dams from the DAMS menu.....	125
Fig. 4.14: Setting Reservoir Fields from the DAMS menu.....	126
Fig. 4.15: Computing Outputs from the DAMS menu.....	129
Fig. 4.16: Updating Values from the DAMS menu.....	130
Fig. 4.17: Input Screen for Initializing Storage.....	131
Fig. 4.18 Generating a Simulation Summary from DAMS menu.....	132
Fig. 5.1: Measured (Target) vs. Simulated (Mflowfit) Daily Flows.....	135
Fig. 5.2: Measured (Target) vs. Simulated (Mflowfit) Monthly Flows.....	135
Fig. 5.3: Measured vs. Simulated Flows at Aoulouz, 1982.....	137
Fig.5.4: Measured vs. Simulated Flows at Immerguen,1992.....	138
Fig. 5.5: Measured vs. Simulated Flows at Amsoul, 1992.....	138
Fig. 5.6: Measured vs. Simulated Flows at Amsoul, 1982.....	139
Fig. 5.7: Measured vs. Simulated Flows at Ait Melloul,1982.....	140
Fig. 5.8: Reservoir Simulation Results, 1990-94.....	141
Fig. 5.9: Variation of Reservoir Storage at Abdel Moumen.....	142
Fig. 5.10: Observed vs. Simulated Evaporation Losses at Abdel Moumen.....	143
Fig. 5.11: Variation of Reservoir Surface Area at Abdel Moumen.....	144
Fig. 5.12: Variation of Reservoir Surface Elevation at Abdel Moumen.....	145

Chapter 1

1.1 PROJECT BACKGROUND

Since the first major river basin simulation by the Harvard Water Program in 1960 (Hufschmidt et al, 1966), tremendous strides have been made in the analysis of water systems. These developments have stemmed from the realization that water is a scarce commodity that must be conserved. Despite the fact that water makes up over 70 percent of the earth's surface, only about two and a half percent of this water is fresh and an even smaller percentage is useable for human consumption and agriculture. With the present rate of population growth, it has been estimated that the demand for water will exceed the available supply within the next 30 years, unless substantial advances are made in the storage and reuse of water. The prospect of this shortage is even more frightening when one takes into account the fact that developing countries are recording the highest population growth rates. These are the countries which have the least resources for exploring new water sources or extracting potable water from non-potable sources.

Development agencies such as the World Bank and the United Nations (UN) are already involved in a number of large scale projects aimed at improving the planning of water resources in developing countries, particularly in Africa. Within the last two years alone the World Bank has embarked on a number of such

projects in Guinea, Guinea Bissau, Niger, Burkina Faso and Nigeria. However, most of this effort has focused on national scale planning. Since most river basins tend to cross national boundaries, the need to coordinate planning and development activities on a basin wide scale becomes apparent. The UN Food and Agriculture Organization (FAO), the United Nation Educational Scientific and Cultural Organization (UNESCO) and the Center for Research in Water Resources (CRWR) at the University of Texas at Austin are undertaking a study called the FAO/UNESCO Water Balance of Africa.

The logical starting point in the adoption of strategies for the prevention of water shortages is an understanding of the behavior of the natural system of water flow. This was the first objective of the Water Balance of Africa Project. The Niger River Basin in West Africa was used as the first case study for the water balance project. The Niger is one of the four main rivers of Africa; the Nile, the Zambesi and the Congo are the other large rivers in the continent. It is an important water resource of the semi-arid Sahel region of West Africa through which it flows. With a drainage area of 2,337,000 km², it provides a typical example of a large river basin crossing a number of national boundaries. Some of the processes defined in the Niger Basin include precipitation, evapotranspiration, surface water, ground water and runoff.

The next step in the research was to represent these processes in such a format as to enable the flow of water through the basin to be simulated. To this end, an ArcView based Geographical Information System (GIS) model was developed using the object oriented programming language, Avenue (Ye, 1996). The model, called GH-Rivers, enables the user to simulate the various natural processes and their interaction with each other. The model has been incorporated into a spatial hydrology analysis system called GISHYDRO97 which contains complementary modules for soil water balance, terrain processing, ground water modeling and water quality analysis. The GH-Rivers model was calibrated to ensure that flows were consistent with the flows observed at runoff stations in the Niger basin. The procedures and methods developed in the Niger have been successfully duplicated in the Souss basin in Morocco. With a drainage area of about 19,000 square kilometers, the Souss is a much smaller river basin than the Niger. It is however the source of water for major irrigation efforts in this arid region of North Africa. Surrounded by the Atlas Mountains on the North and the Anti-Atlas Mountains on the South, the basin extends for about 200 kilometers inland from the Atlantic Ocean. The location of the river basin in a flat plain surrounded by steep mountains implies that rainfall quickly becomes runoff and concentrates in the low lying regions in the center of the basin. This water subsequently seeps in to the underlying subsurface aquifer. Agriculture has thrived in this area by exploiting the ground water resources offered by the region.

Galleries have been dug into the ground water table and the water allowed to flow by gravity over the land surface to irrigate the farms. Over the past 50 years, increased demand has led to increased pumping of water from the aquifer leading to a decline of the water table. Estimates of this decline run as high as 100 meters in some places. There is a need for surface and ground water models to describe the horizontal and vertical movement of water to facilitate better quantification and planning of the water resources of the basin.

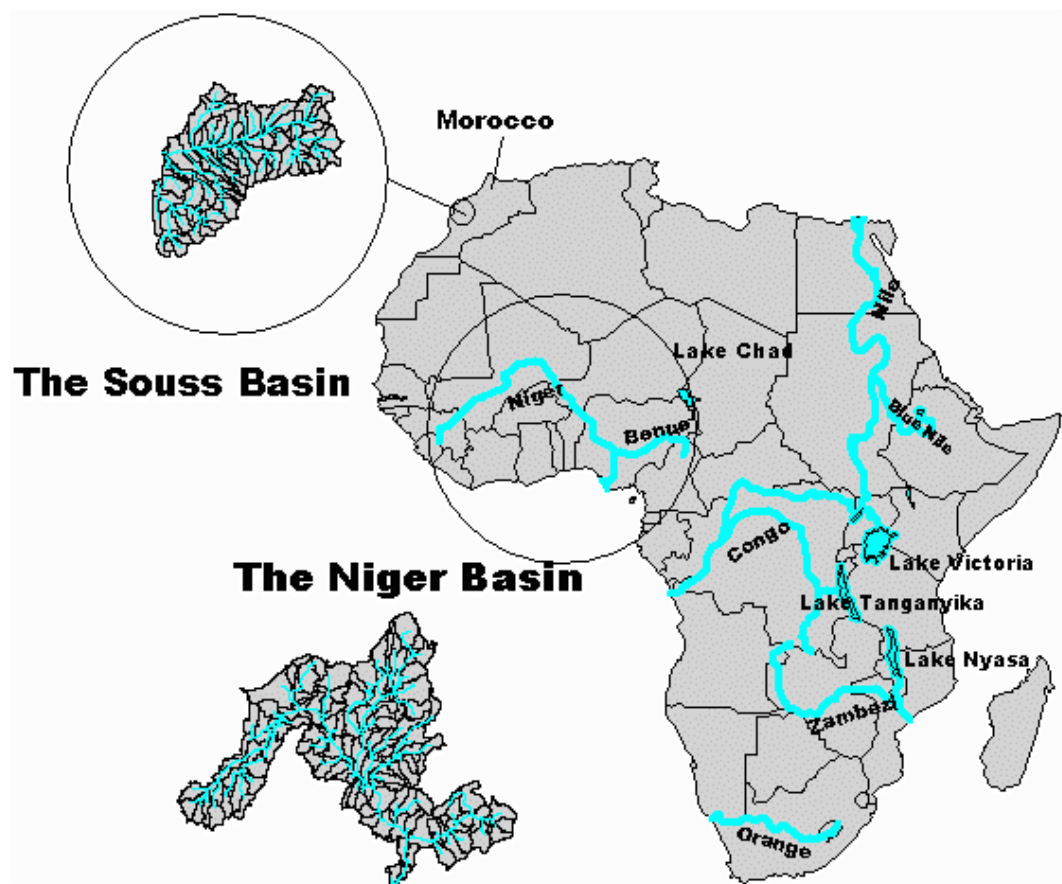


Fig. 1.1: Map of the Major Rivers and Lakes of Africa

The final step in the research is to develop the model into a tool that can be used to answer some of the questions frequently asked by water resource planners. In developing water resource systems, there is often a need to determine what are the likely demands on the system, and to assess the ability of the system to meet these demands. Models which provide insight into the consequences to other users of allocating a quantity of water to one user are required for such analyses. This study is meant to be the starting point in the development of models for the examination of the effects of control strategies. The study focuses on reservoir planning because of the importance of reservoirs in regulating flow in rivers. Models which simulate the effects of reservoirs can provide a basis for planning water development activities. The Souss basin in Morocco contains three major reservoirs which serve a variety of uses. Two of these reservoirs are located on the same river reach and are operated in concert with each other. Hence the Souss basin provides an excellent setting to study the effects of incorporating reservoirs into a natural river system.

Recent developments in Geographical Information Systems (GIS) and map based modeling in particular have made it a versatile tool in the management of natural resources. GIS based systems have been used extensively in fields such as forestry, land use and wildlife management. The adoption of GIS as the water

resource inventory and planning tool is therefore in line with the goal of integrated resource management, espoused in the UN Agenda 21 (UNCED, 1992).

1.2 MOTIVATION

The availability of potable water is one of the most important determinants of the quality of life in many communities in developing countries. With limited resources to develop alternative water supplies, these communities often depend on the water resources that are closest to them. Local authorities and other development agencies consequently invest considerable resources into developing such water supplies to meet the needs of these communities. This may result in a situation where numerous projects, located at different points along a river, are relying on water from the same source. With limited resources to invest in the construction of complex models, important decisions on the availability of water to sustain projects are often based on grossly simplified representations of the river system. This has been identified by the World Bank (Sharma Etal, 1996) as one of the main reasons for the failure of water resource development projects in developing countries. Consequently, there is a need for inexpensive tools that enable planners to accurately quantify the available water supply and its ability to meet the competing demands of projects.

Computer simulation models have been used extensively for analyzing river systems. However, many of these models are limited in application to the particular river basins and require the collection of large amounts of data. The construction of such models and databases are expensive undertakings requiring highly skilled personnel. Developing countries often do not have the resources and personnel to develop such models and yet it is these same countries that have the greatest need for models that enable them to better manage their natural resources. The establishment of global databases of geographic, climatic and other physical parameters is one of the exciting, new developments in Geographic Information Systems. Modelers no longer have to construct their own data bases from scratch in order to build a model of a river basin. They can begin building the models with data from already established sources. The GISHYDRO97 Spatial Hydrology Analysis System developed at the Center for Research in Water Resources, the University of Texas at Austin, enables a modeler to develop a model of a river basin using data from some of these established sources. The map based simulation approach presented in GISHYDRO97 is a major step forward in the modeling of a river basin because it reduces considerably the time and input data required to construct a model. The technology in this area of map based modeling is still evolving but some of the potential benefits such as cost savings on database creation are already becoming evident. This study aims to document the development of a river basin model using the GH-Rivers model and some of the

data from the established global databases. It is hoped that the approach will provide a quick and inexpensive means of analyzing the effects of water resource planning decisions in river basins.

1.3 OBJECTIVES

A number of the tools presented in GISHYDRO97 are required for performing many of the tasks involved in producing a simulation model. These tools include surface reconditioning models, watershed delineation programs and surface water simulation models. One of the key objectives of this study is to document the use of some of these GIS tools for developing a surface water simulation model in a typical river basin. The data requirements, data sources, steps in processing this data into a hydrologic model and the calibration of the model are also documented. Based on the results of the simulation, inferences are made as to the type of information that such a model can reliably provide.

The next key objective is the incorporation of man made structures into the simulation of the river basin. In most prior work in map based surface water simulation, the operation of reservoirs has not been incorporated into the simulation. Reservoirs have been simulated separately, with only symbolic lines (not the actual river arcs) linking them. Flow in the Souss basin is highly regulated by dams along major reaches in the basin. The availability of historical operational

data as well as volume-elevation and surface area-elevation curves for the reservoirs in basin made it possible to incorporate the regulation data into the simulated flows. Consequently, the Souss basin provides an excellent setting to incorporate the effects of dams and reservoir into a model of a natural river system.

The final objective of the study is to examine the utility of the model as a decision support tool. The critical test in meeting this objective is the ability of the model to predict the amount of water available at various locations in the basin under a variety of reservoir operation scenarios. For a model to meet this criterion, it has to provide the user with the flexibility to modify the components of the river system as well as the operation patterns of individual reservoirs in the system. The reservoir simulation portion of the model has to keep track of water levels in the reservoirs while still providing a mechanism for user interaction in making water allocations from the reservoirs. The development of such modeling capabilities is considered an important step in making the map based surface water simulation model useful as a planning tool, not just a research exercise.

1.4 METHODOLOGY

There has been a lot of research in the area of surface water simulation and the key developments in this field are highlighted in chapter 2 of this report. Reservoir simulation techniques and models that have been developed are also reviewed as a basis for developing the model for this study. Other research efforts involving the use of GIS as a planning tool in the field of water resource and environmental engineering are also highlighted in chapter 2.

The map based surface water simulation model, GH-Rivers developed at the Center for Research in Water Resources, the University of Texas at Austin, is used in this study. The model was developed using the object-oriented programming language, Avenue, supported by the GIS software, ArcView. The software uses a graphical user interface to display coverages which are maps of geographically referenced objects. These objects, topologically represented as polygons, lines and points, are linked to databases containing related information. The process of creating these coverages and the associated databases, illustrated in Figure 1.2, is discussed further in chapter 3 of this report. The chapter also details the process of developing rainfall, runoff and reservoir operation time series.

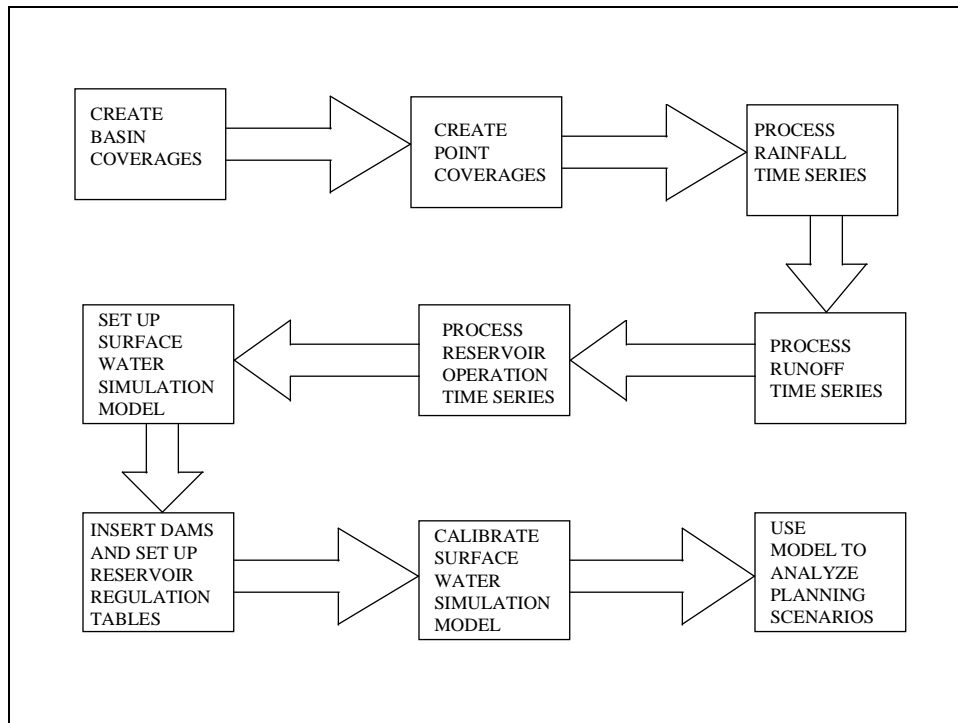


Fig. 1.2: Developing a Map based Surface Water Simulation Model

The assembly of these components into a hydrologic model is the next step in the process. The model was calibrated to generate flows comparable to the observed runoff data. This was followed by the construction of the reservoir simulation which enabled the various operation scenarios of the reservoirs coverage to be simulated. This involved modifying the existing simulation routine in the program. Chapter 4 outlines this procedure and demonstrates how the resulting model can be used as a planning tool.

The results of the simulation in the Souss basin and a summary of findings are presented in Chapters 5. An assessment of the method and its limitations and advantages are presented in the final chapter. The programs developed during this study are presented in the appendices. The programs include operating instructions to enable subsequent users to duplicate the process.

1.5 CONTRIBUTION TO KNOWLEDGE

This research was undertaken with the intention of consolidating the knowledge acquired in GIS and map based surface water simulation into a single procedure for developing a model of a river basin that includes reservoirs. It is hoped that the documentation of this procedure will provide modelers and planners with a handy reference that could be used for developing similar models in other river basins. The reservoir simulation undertaken in this study represents an important first step in transferring the knowledge acquired from research institutions to the working environment where decisions have to be made. It provides a means of modeling not just the natural system as it exists but also the effects of changing those variables over which we have control.

Chapter 2: Literature Review

2.1 INTRODUCTION

A review of research in the three major aspects of this study was undertaken. They include surface water simulation models for generating flows from rainfall data, reservoir simulation models for predicting storage levels and the use of GIS as a planning tool.

2.2 SURFACE WATER SIMULATION

A number of models for simulating the rainfall-runoff process have been developed in recent years. These models can be classified by their output as flood routing models, continuous stream flow simulation models, hydraulic routing models or water quality models (Maidment,1993). Continuous stream flow models are the most appropriate class for analyzing the effects of water allocation in a river basin. The most commonly used of this class of models is the distributed parameter type which takes into account the spatial variability of land surface characteristics influencing the flow of water. Runoff from any watershed in a river basin is predicted based on rainfall patterns, watershed characteristics and hydrologic response. The hydrologic response may be expressed either in terms of mathematical equations or as simple lumped coefficients (Kouwen et al,1993). The

main limitation of distributed parameter models is that they require large amounts of data to define the watershed characteristics and hydrologic response (Ross et al,1993). The collection of such data can be an expensive venture. Parameters collected for a single location in a basin may subsequently be extrapolated for the whole basin, compromising the reliability of the model. A map-based surface water modeling approach enables hydrologic response to be determined by an optimization process (Ye et al,1996). The flow direction is determined from land surface elevation considerations but the quantity and timing of flows are determined by lumped parameters. The parameters are optimized to ensure consistency between simulated and measured runoff. This approach allows a basin to be simulated with limited amounts of soil and other hydrologic response data.

2.3 RESERVOIR SIMULATION

The problem of water availability and distribution in reservoirs has been the subject of many research efforts. The main facets of the problem include the prediction of inflows, the estimation of demand, the allocations of water to meet the conflicting demands of various users, and the development of operating policies (Loucks et al,1981).

Stochastic and statistical methods have been used to analyze runoff data and subsequently to forecast reservoir inflows (Takeuchi et al,1995). The generation of such inflow is only necessary when the reservoirs in a system are removed and analyzed in isolation. The traditional approach to reservoir simulation involves representing the reservoirs by a schematic diagram with a series of symbols and lines to depict structural elements and river sections linking reservoirs. The whole model has to be reconstructed every time an element (such as a reservoir) is removed or added, in order to keep the interconnections current. When the reservoirs are incorporated into the map based model as was the case in this study, reservoir inflows are generated by the surface water simulation component. The location of structural elements can be changed without having to reconstruct the entire model.

Conjunctive water use models have been used to address the problem of variable demand (Teegarden,1994). These models often incorporate both optimization and simulation techniques to determine optimal values of decision variables. Water management decisions are made on the basis of a series of state variables including hydraulic head, streamflow, and aquifer-surface water interchange. The main limitation of this approach is that a model developed in one basin cannot be transferred to another. This is because the relationship between state variables and demand may vary from basin to basin. The approach also has

the in-built assumption that demand is based solely on hydrologic conditions. When water is used in irrigation farming, for example, the availability of labor and market forces may also influence demand. Conjunctive use models may not allow the user to take these factors into account. The approach used in this study, allows the planner to choose between using the historical demand data and supplying a new set of demand data for the simulation.

The problem of making allocations to meet the need of multiple users is a classical water resource management problem. The use of optimization techniques in finding optimal allocations is well documented (Yeh, 1985; Wurbs, 1993). These methods often result in single correct values of allocations. Grey programming methods have also been developed to allow decision makers a little more leeway in making allocations by providing a range of feasible values instead of single correct value (Yeh, 1996). Objective functions used in an optimization model have in-built value judgments which may vary from region to region. It would hence be unreasonable to treat these solutions as infallible. Another limitation is that optimization models are usually constructed on a case by case basis and cannot be generalized to serve a variety of river basins. When simulation is used instead of optimization, the objective is to examine the effects of selected scenarios rather than to find a single correct solution.

Operating policies are often derived by running multiple scenarios of the resource allocation problem and adopting policies that optimize the desired benefits. These scenarios may include system variations, flow variations and demand variations. Models for deriving operation policies often use daily or weekly time steps (Hsu et al,1995). The most commonly used of these are the HEC-3 and HEC-5 models developed by the US Army Corps of Engineers Hydrologic Engineering Center.

2.4 GIS AS A PLANNING TOOL

A number of research efforts have already explored the use of GIS for decision support in water resource management and planning. Many of these efforts involve the integration of two or more models with the GIS being used for data management and spatial analysis (Mattingly,1994). Integrating optimization models with GIS for spatial analysis, is another recent development in this field. The integration of GIS and Decision Support Systems (DSS) has resulted in the evolution of Spatial Decision Support Systems (SDSS) (Walsh,1993; Kilgore et al,1994). These systems enable the planner to analyze various water resource management scenarios. A few systems such as the PEGASUS (Planning Environment for Generation and Analysis of Spatial Urban Systems) enable water resources to be incorporated into the wider urban planning process (Kim, 1996).

Models employing GIS as a hydrologic modeling tool mostly operate in a raster environment. Raster GIS models have been used to quantify water resources in regional (Reed et al,1997), basins (Lullwitz et al,1995) and local studies (Conor et al,1993). The surface water simulation model developed at the University of Texas at Austin appears to be the only documented example of a model operating wholly within a map based, vector GIS environment (Ye et al, 1996). With the inclusion of a reservoir simulation model in the hydrologic model, water allocation and other planning activities can be simulated and their effects across the entire river basin observed.

Chapter 3: Data Processing for the Souss Basin

For a GIS model to be useful, it must be accompanied by a supporting database. This chapter outlines the procedures followed in developing the database for the Souss basin hydrologic model. The programming language Avenue used in the development and customization of the GIS software ArcView is an object-oriented language. Coverages and tables containing data from the study area are defined as objects and are manipulated by programs called scripts. The objects used in the database must conform to a specified format to allow for easy manipulation in the scripts. The processing of the Souss basin coverages is discussed along with issues encountered in the establishment of the three major data components of the database, namely the rainfall, runoff, and reservoir tables.

3.1 PREPARING THE DATA

One of the key factors determining the reliability of any simulation effort is the quality of the data used in defining the system. However, preparing a quality database can be a very costly and tedious undertaking. The simulation program used in this study requires very little system definition data thus considerably reducing the database construction task. The data must however undergo some format changes before it can be used in the program. The input data and preprocessing steps required by the program are outlined in the following sections.

But prior to reviewing these steps, the structure of the database created for the Souss basin study is presented to highlight the location of various components. All the data and programs presented in this database are stored on the **Souss Basin Model** CD-ROM available from the Center for Research in Water Resources at the University of Texas at Austin. The contents of this CD are given in Figure 3.1.

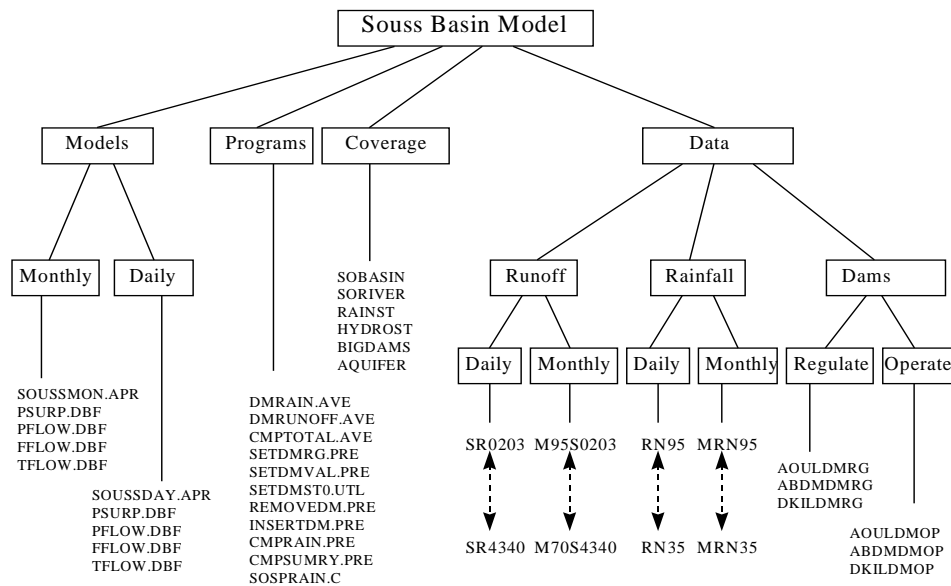


Fig. 3.1: Souss Database Structure

The **Models** directory contains the two sub-directories, **Monthly** and **Daily**. These sub-directories contain the project files (*.apr) which enable ArcView to run the monthly and daily simulation models, respectively. They also contain the time series tables, control files and coverages created by the preprocessor in GH-Rivers to store the results of simulation operations. Control

files are text files which are used to store inputs required by various simulation programs. They enable the developer of a program to provide standard inputs for many model processes, thus reducing the number of inputs that must be specified by subsequent users. The preprocessing procedure for the GH-Rivers model is described in section 4.1 of this report.

The Programs directory contains scripts for the ArcView based data processing programs developed in this research. These include programs for converting daily rainfall data into total monthly rainfall data (**Dmrain.ave**) and daily runoff into mean monthly runoff rates (**Dmrunoff.ave**). Another data processing program, **Cmprain.pre**, interpolates for missing records in the rainfall time series by assigning the value of rainfall recorded at the nearest rain gage to missing records. One program written in the programming language C is also included in the directory. This program, called **sosprain.c**, converts data stored in two dimensional tables to one dimensional time series. A series of programs used for setting preprocessing reservoir simulation data are also stored in this directory. They include programs to create reservoir regulation tables (**Setdmrg.pre**), alter monthly allocations (**Setdmval.pre**) and initialize reservoir storage (**Setdmst0.utl**). Other programs in this directory include **Inserdm.pre** and **Removedm.pre** which are used for inserting and removing dams from the surface water model, **Cmptotal.ave** which computes net demand or release for reservoirs

with multiple uses and **Cmpsumry.pre** which can be used to generate a summary after each reservoir simulation run. The **Programs** directory also contains a **data dictionary**, detailing the location and contents of all the files that make up the Souss model. The data dictionary and the programs listed in this directory are presented in appendices of this report.

The **Coverage** directory contains the coverages generated from external data sources. These coverages are different from those created by **GH-Rivers** during preprocessing. (The latter are stored in the respective monthly and daily directories under which they are generated during preprocessing). The sources of the data used and the procedures followed in creating these coverages are described in section 3.2 of this report. The coverages in this directory include the Souss basin watershed polygon and boundary line coverage, **Sobasin**, the river network line coverage, **Soriver**, and the aquifer polygon coverage, **Aquifer**. The directory also contains point coverages of the rain gages locations, **Rainst**, the runoff gauging stations, **Hydrost**, and the location of the major reservoirs in the basin, **Bigdams**.

The **Data** directory on the **Souss Basin Model** CD-ROM contains three sub-directories, namely **Runoff**, **Rainfall** and **Dams**. The **Runoff** directory

contains daily and mean monthly runoff data recorded between 1970 and 1995, at 6 runoff stations within the Souss basin. The **Rainfall** directory contains daily and total monthly rainfall from 48 rain gages located around the basin between 1935 and 1995. The rainfall data in this directory has been interpolated to remove missing records. The **Dams** directory contains two sub-directories, **Regulate** and **Operate**. The **Regulate** sub-directory contains data for defining the reservoir regulation curves for the reservoirs at Aoulouz, **Aouldmrg**, Abdel Moumen, **Abdmdmrg**, and Dhkila, **Dkildmrg**, in the Souss basin. The **Operate** sub-directory similarly contains reservoir operation data for the reservoirs at Aoulouz, **Aouldmop**, Abdel Moumen, **Abdmdmop**, and Dhkila, **Dkildmop**. These files contain historical inflow, storage and water allocation data covering the entire period of operation of the respective reservoirs.

3.2 DEVELOPING RIVER BASIN COVERAGES

The coverages used in this study can be classified into two main categories. The first category, which will be referred to as **basin coverages**, consist of polygon and line coverages defining the flow path and watershed characteristics of the river basin. Polygon features defined in the Souss basin include watershed basins and the aquifer recharge zone. Line features include streams and the basin boundary outline. The second category of coverages consist of point coverages

used for indicating the location of **data** or **control points**. These are features which play important roles in the simulation model but whose physical dimensions are too small to be explicitly represented on the map. Point coverages were used in the Souss basin to represent the location of rain gages, runoff stations and dams. The approaches used in developing these two categories of coverages are describe below.

Basin coverages for the Souss basin were developed from 30'' digital elevation models (DEM) developed by the United States Geological Study (USGS) as part of its GTOPO30 project. A DEM may be defined as a topographic surface arranged in a data file as a set of regularly-spaced x, y and z coordinates where z represents the elevation (ESRI, 1992). These 30'' DEMs have been prepared for the entire world. This is the equivalent of covering the land surface of the earth with a mesh of cells about one kilometer by one kilometer in size. The elevation in each of these cells is presented as a single value in meters above sea level. While digital elevation data may be represented in various ways, the format adopted by the USGS is widely recognized as the industry standard. These DEMs are available for down loading, without cost, from the USGS website at the following address:

“<http://edcwww.cr.usgs.gov/landdaac/gtopo30/gtopo30.html>”

The procedure outlined below was used in downloading the DEMs for the portion of Africa containing the Souss basin. The commands were run from a UNIX based system. A directory was created for storing the DEM by typing :

\$mkdir demdata

from the UNIX command prompt. The USGS site was accessed with an Internet browser. An image appears with the world divided into 27 tiles, each covering 40⁰ of longitude and 60⁰ of latitude. The DEM can be previewed by clicking on the image of the tile to be downloaded. The continent of Africa is divided into 4 tiles. The north-western portion of Africa where the Souss basin is located, is stored in tile number W020N40. This is a compressed UNIX tar file containing 16.94 Megabytes of data. The spatial extent of this tile covers the region defined by the coordinates 20°W to 20°E and 40°N to 20°S. Clicking on the hyperlinked text of the tile number initiates the transfer of the compressed file. A prompt comes up for the user to specify the target directory in which the file is to be stored. The **demdata** directory earlier created is specified as the target directory.

The downloaded DEM was uncompressed, using the gunzip utility by typing :

\$gunzip afr_dem.dem W020N40.BIL.GZ.

from the UNIX command prompt. The **.dem** file extension is required to facilitate subsequent conversion of the DEM into a format readable by the Grid module of

Arc/Info. To convert the DEM into grid format, open Arc/Info with the command

: \$arc

The program displays a message indicating that it has opened Arc/Info. Typing :

Arc : grid

opens the grid module of Arc/Info. A graphics display window is opened with :

Grid: display 9999

The DEM as it is acquired from the USGS is an image file. It needs to be converted to a grid before any spatial analysis can be performed on it. This conversion is done with :

Grid : gridimage afr_dem.dem afr_dem

The grid of the whole tile can be viewed in the display window by typing :

Grid : mape afr_dem

to define the map extent followed by :

Grid : gridpaint afr_dem value linear nowrap grey

The grid created by the processes above includes much of North-West Africa . The Souss basin is contained within the area bounded by the coordinates This study area is isolated by redefining the extent of the grid with the command :

Grid : setwindow *

The resulting cursor is used to reset an analysis window by first clicking in the top left hand side of the area and dragging the cursor to the bottom right hand side of

the study area before clicking again. The study area for the Souss basin is defined as the area bounded by latitudes 29 ° and 31 ° and longitudes -7.5 ° and -10°.

Grid : sos_dem = afrdem

creates a new grid, sos_dem, containing only the portion of the afrdem within the defined box. This new grid, while smaller than the original grid, still contains portions of the Atlantic Ocean and Mediterranean Sea which are not required for the surface water model. The elevation of offshore cells are differentiated from land surface elevation by assigning high default values to them. Such cells can be excluded from the Digital Elevation Model with the conditional function

Grid: soscont = con(sos_dem <> 55537, sos_dem)

Now, the grid containing the Souss basin without the rest of the area in the tile or the sea is displayed on a gray scale with the following sequence of commands :

Grid : clear

Grid : mape soscont

Grid : gridpaint soscont value linear nowrap gray

In order for the DEM to be used with other data from the Souss basin, it must be projected from a geographic coordinate system into a Lambert Conformal Conic system, the standard projection in Morocco. Using this projection ensured that existing maps developed in Morocco could subsequently be used in the model. Maps of vegetation, land use and population distribution may be required for impact assessments and other planning purposes. Adopting a different map

projection can render such existing maps unusable because of relative accuracy problems associated with going from one projection to another. The parameters for this projection are stored in a text file called **sosprj** as shown in table 3.1. The return button is pressed after the end statement to ensure that the end command is executed.

input
projection geographic
units dd
datum nad83
parameters
output
projection lambert
units meters
datum nad83
parameters
29 5 53
31 16 30
-5 24 00
29 42 00
5000000
300000.0
end

Table 3.1: Sosprj projection file

The grid **soscont** is projected to **sodempj** using the projection file **sosprj** by typing the following at the arc prompt:

arc: project grid soscont sodempj sosprj

At the time the coverages for the Souss basin were processed, the raster data analysis tool was not yet available in the ArcView software. All the data

processing had to be done in the raster environment of the Arc/Info module, Grid. This is a command driven tool which can be somewhat cumbersome to use because of the strict command syntax that must be observed. With the release of ArcView3 and the Spatial Analyst, the processing of DEMs can now be done in a much more friendly Windows environment. The GISHYDRO97 Terrain module presents ArcView-based tools for accomplishing this task.

The original DEM contains low elevation points known as pits. These pits must be removed if the flow of water over the land surface is to be simulated, effectively. The pits are removed by defining a maximum pit depth over the study area. All cells with depths greater than the maximum depth are filled in. The command for performing this task is as follows:

Grid: fill sodempl sodemfl SINK 800 #

where 800 is the maximum pit depth in meters, and sodempl and sodemfl are the input and output grids, respectively.

The next task involves defining the direction in which incident rainfall would run off the land surface. Since water is most likely to flow in the direction of the steepest slope, the slope between a cell and its 8 neighboring cells is used to define the flow direction over the land surface. The algorithm used defines flow direction from a particular grid cell as the steepest path from its centroid to that of

a neighboring cell. It computes the slope to centroid of its eight neighboring cells and defines flow in the direction with the steepest downward slope. A grid with the flow directions in the basin defined can be created with the command:

Grid: `sodemfd = flowdirection (sodemfl)`

The resulting flowdirection grid has cell values assigned depending on which of the directions of an eight point compass the cell discharges in. Table 3.2 shows the flow direction grid cell value for the eight compass directions.

<u>Flow Direction</u>	<u>Cell Value</u>
East	1
South-East	2
South	4
South-West	8
West	16
North-West	32
North	64
North- East	128

Table 3.2: Cell Values in the Flow Direction Grid

The next step is to compute a flow accumulation grid. This is a grid of the number of upstream cells contributing flow to each cell. It is computed from the flow direction grid with the command:

Grid: `sodemac = flowaccumulation (sodemfd)`

With the flow accumulation grid in hand, it is now possible to identify areas in the basin where water is likely to accumulate. It is also possible to

determine where a river is most likely to flow since a certain amount of water is required to create a river. A continuous string of high accumulation cells can be defined as a river. There is however a need to make a distinction between small streams of little significance and larger rivers. This is done by defining a minimum threshold of the number of upstream cells that a cell must have in order to be classified as a stream. This threshold may vary depending on the complexity of the stream network required. For the Souss basin study, a threshold of 100 was used. This means that cells with a minimum drainage area of 100 km² are classified as falling within the river channel. The command for delineating the stream network with a threshold is :

Grid: sodemsn = con (sodemac > 100, 1)

The resulting grid contains cell values of 1 for cells with a drainage area greater than the minimum threshold and cell values of 0 for all other cells. It contains a single stream network with several tributaries all joining the main river. In order to determine the areas that are contributing flow to individual river sections, this stream network must be broken down into its component arcs. The command :

Grid: sostlnk = streamlink (sodemsn, sodemfd)

separates the stream network into the component arcs using the flow direction grid to locate intersection points. The arcs in each river reach must now be assigned the same value so that they can be identified with the other cells in the reach. The command:

Grid: sostmax = zonalmax (sostlnk, sodemac)

assigns the maximum value of flow accumulation in each river reach to all the other cells in that reach. The outlet cells in the reach are now identified as the cells for which the cell value in the zonal maximum grid, sostmax, is equal to the flow accumulation grid and which also lies within the same river reach as defined by the stream link grid. This is done by typing:

Grid: sostout = con (sostmax == sodemac, sostlnk)

The portion of land draining through the outlet points in **sostout** can be delineated using the command:

Grid: sodemws = watershed (sodemfd, sostout)

This produces a grid in which there is a watershed zone associated with each outlet point. This zone of cells represents the area of land draining through the outlet point. Consequently, whenever rain falls on that watershed, any water that becomes surface runoff will drain through that outlet. The processing of the DEM in the grid environment is now complete. The Grid module can now be shut down by typing:

Grid: quit

The watershed grid, **sodemws** is converted into a polygon coverage for use in the ArcView based model from the arc command prompt:

arc: polygrid sodemws sobasin

The watershed boundary arc coverage can be constructed from the polygon coverage using the build command.

arc: build sobasin

A label point coverage is also produced containing one label point per polygon. These label points are used for storing information about individual polygon (polygon attributes) in Arc/Info. However, they are not required in ArcView which stores attribute information in a separate table called the feature attribute table. Each feature in the coverage is linked to the corresponding attribute information in the feature attribute table by a key field common to both the coverage and the table. The key field for basin coverages in the Souss model is the grid-code.

The stream network is created by converting the zonal grid, **sostmax**, into a coverage and then building it as a line coverage using:

arc: linegrid sostmax soriver

arc: build soriver arc

The delineation of the watersheds in Souss basin using the process described above produced a coverage with 114 polygons and associated river arcs as shown in Figure 3.2.

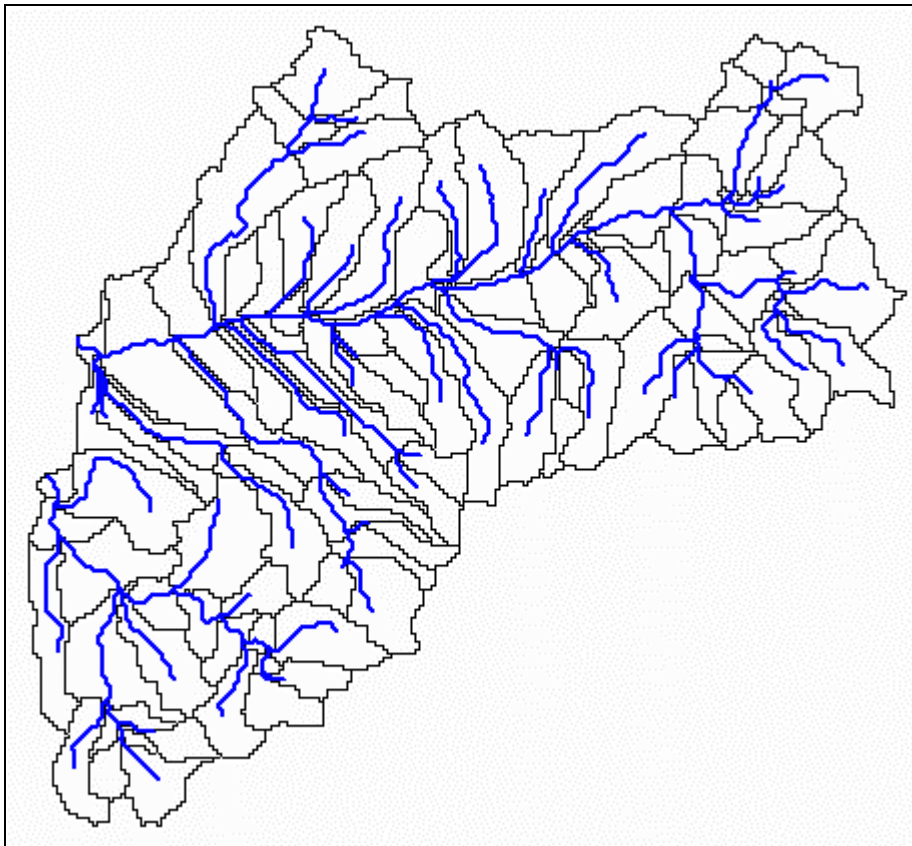


Fig. 3.2: Delineated Streams and Watersheds

The watersheds were of comparable size but there were a few tiny arcs which can result in single-cell, watershed polygons. If single-cell polygons are encountered, the watershed and river network coverages must be edited to remove them before the coverages can be used in the simulation model. Coverages may also be edited to reduce the number of watersheds and consequently, the model calibration effort. Instructions for editing coverages in the Arc/Info module, **ArcEdit**, operating from a UNIX environment, are provided below. The

commands described here are those required for performing some of the basic editing tasks encountered in editing river and watershed coverages. Due to the interactive nature of the edit environment offered by ArcEdit, the instructions included here cannot be an exact duplicate of those used in editing the Souss basin coverages. These instructions have compiled to highlight the key commands in ArcEdit and to explain how they can be used in editing basin coverages.

Begin the ArcEdit session by logging onto the computer system and opening up a console window. From the UNIX command prompt, type:

\$arc

to start Arc/Info. Begin an Arc Edit session from the resulting arc prompt by typing:

Arc: arcedit

At the arcedit prompt, type:

Arcedit: display 9999

to open up a display window. The coordinate system defined in the river network coverage, **soriver**, may be used to define the map extent in the display window with the command:

Arcedit: mape soriver

This is however not necessary as the next command sets the map extent to the boundaries of the **soriver** coverage by default. The command:

Arcedit: ec SORIVER

specifies the **soriver** as the coverage to be edited and sets the map extent if there is no extent defined. The type of features to be made drawn in the display window must be specified by setting the draw environment to arc using:

Arcedit: de arc

The edit coverage can now be viewed in the display window by typing:

Arcedit: draw

The default color of the edit feature is white. This can be changed with the command:

Arcedit: calc &symbol = N

where N is the number of the color to be used. For example, using N = 2 changes the default color to yellow. It is also a good idea to have the watershed coverage or any other coverage displayed in the background. The command:

Arcedit: bc sobasin 5

allows the watershed coverage, **sobasin**, to be displayed as a backcoverage with the color number 5, a light blue. As with the edit coverage, the class of feature to be displayed in the background must be specified. The command:

Arcedit: be arc poly

allows both arc and polygons to be displayed in the back environment. Make the newly defined backcoverage visible in the display window with:

Arcedit: draw

Specify the type of feature to be edited in the edit coverage, **soriver**, with:

Arcedit: ef arc

The set up of the edit environment is now complete and changes can now be made to the coverage. Select an arc that is to be deleted by typing:

Arcedit: sel

A cross-hair feature selection tool appears in the display which can be used to select the arc. The selected arc becomes yellow. Repeat the select command if the wrong arc (or no arc) is selected. Multiple arcs can be selected with:

Arcedit: select many

It is a good idea to zoom in a little closer before selecting the arc. The **zoom** command in the menu bar of the display window allows the user to do this.

Delete the arc or arcs by typing:

Arcedit: delete

Deleted arcs can be restored to the coverage by typing:

Arcedit: oops

The **oops** command can be repeated as many times as desired to restore prior edits one at a time. There is also the option of leaving Arc Edit without saving the edits. This option ensures that all edits performed during the session are discarded.

It is sometimes necessary to split an arc into two to enable portions or it to be deleted. Select the arc using the select command described earlier. Split the arc into two by typing:

Arcedit: split

and clicking on the location along the arc where the node is to be inserted. A node is insert at the selected point and two independent arcs are formed. The unwanted arc can now be selected and deleted. It may also be necessary to extend arcs which may be isolated from the rest of the river network to ensure that the network remains connected. Select dangling arcs using the **select** command and type:

Arcedit: extend

Click on two locations in the display window to indicate the distance over which the arc should be extended. The selected arc will only be extended if there is another arc within the specified distance. It is important to keep the edit distance small since both ends of the selected arc will be extended. If both ends of the selected arc are close to other arcs, it may be necessary to move one node of the selected arc a little closer to the nearest network arc. To do this, set the edit feature to node with:

Arcedit: ef node

Move the nodes by typing:

Arcedit: move

An indexed menu is presented with index numbers specifying the actions to be performed. Choose numbers from the menu to specify the node to be moved and the destination to move it to. Now, set the edit feature to arc again and try the

extend command again. When the editing is complete, save the changes to the coverage by typing:

Arcedit: save soriver

and exit Arc Edit with the command:

Arcedit: quit

From the arc prompt, restore topology in the edited coverage with the command:

Arc: build soriver arc

The edited coverage is now ready for use in the simulation model.

The procedure describe above can also be used for editing the watershed polygon coverage, **sobasin**. The edit feature is set to arc as before since polygons are described in Arc/Info topology as arcs with labels in them. The edited stream network coverage is used as the backcoverage so that unwanted polygons, those with no arcs in them, can be identified and removed. Another important difference to be addressed is that reducing the number of polygons without a corresponding reduction in the number of labels results in some polygons having multiple labels. Extra labels can be detected and removed at the end of the edit session but polygon topology must be reestablished in the edited coverage before this can be done. Save the session and exit Arc Edit as before. Recreate polygon topology in the coverage by typing:

Arc: build sobasin poly

The **build** command can only be used to establish topology if there are no intersecting arcs or arcs running too close to each other. If any intersections are detected, an error message is displayed indicating that polygon topology could not be created. In such instances, the **clean** command must be used to remove intersections and reconstruct polygons:

Arc: clean sobasin sobasin2 0.0001 0.0001

This results in a new polygon coverage called **sobasin2** with polygon topology established. Creating a back up copy like this is highly recommended because ‘cleaning’ coverages can sometimes have undesirable consequences, such as the creation of sliver polygons and the loss of attributes. After polygon topology has been reestablished, check for multiple labels by typing:

Arc: labelerrors

Make a list of the identification numbers of the polygons with errors and restart the Arc Edit session. After opening the display and the edit coverage, set the draw environment to polygons, arcs and labels using:

Arcedit: de poly arc label

and the edit feature to label with:

Arcedit: ef label

Select the extra labels by specifying the label field numbers obtained from the labelerrors command by typing:

Arcedit: sel sobasin2-id = 45

where **sobasin2-id** is the name of the identification field (usually the coverage-id) and 45 is the label number. The command:

Arcedit: delete

removes the selected labels. Save the changes and exit Arc Edit, and build the coverage again.

The second category of coverages, the point coverages, are used to mark the location of control points and structures. In the Souss model, point features are used to mark the location of rain gauges, runoff stations and dams. A separate coverage is developed for each of these feature types. Data requirements for developing a point coverage include a file containing the geographic coordinates of each feature and a point identification field such as a station number. For the Souss basin model, station identification numbers assigned by the Direction Generale de L’Hydraulique (DGH) are used in developing the rain gage and runoff coverages. The DHG is the agency responsible for the operation of the stations or control structures in Morocco. Using station identification numbers defined by them ensures that other associated data such as rainfall and runoff records can be linked to the corresponding station. The station number is used as the key field for such data linking. A new set of numbers were assigned for the reservoirs as there were no predefined identification numbers. Additional data such as station name, elevation and date of creation were added to the attribute table in ArcView. This

makes the point coverages more informative and facilitates subsequent querying. The procedure for developing a point coverage is outlined below using the runoff stations as an example. A similar procedure is used in developing the other point coverages. A data file containing the locations of the runoff stations is created in a text editor and saved under the file name **hydrost.dat**, as shown in Table 3.3.

1	-9.155	30.753
2	-9.071	30.844
3	-9.061	30.702
4	-8.018	30.600
5	-9.492	30.362
6	-8.155	30.700
7	-8.897	30.433
8	-9.191	29.728
9	-9.510	29.616
10	-8.017	30.727
end		

Table 3.3: The Hydrost.dat File of Runoff Station Coordinates

The data file is divided into three columns, each separated by a single space. The first column contains an index running from 1 to 10 (or the appropriate number of stations). This index is used to mark the beginning of data for a new point in the data file. The second and third columns contain the respective longitudes and latitudes (in decimal degrees) of the runoff stations. While runoff data made available for this study included only 6 runoff stations, there were a

total of 10 runoff stations in the coverage. The additional runoff stations were included to ensure that subsequent studies could use runoff data from any of the other stations in the basin without having to develop a new coverage. When the text file is created, the 'end' statement must be followed by a click on return key. This ensures that it is executed by the program when the coverage is being generated. It indicates to the program that data input is complete.

Additional attributes can be added to a coverage to make it more informative. A file containing additional attributes of the runoff stations is created and stored under the file name, stattrib.txt. The fields in this file include the name of the station (stname) and the station id number (IRE). The text file, stattrib.txt, is shown in Table 3.4.

HYDROST_ID	STNAME	IRE
1	AGUENZA	0597
2	AMSOUL	0594
3	-	-
4	IMMERGUEN	0642
5	AIT MELLOUL	4340
6	AOULOUZ	0203
7	-	-
8	-	-
9	-	-
10	IBERGNATEN	0858

Table 3.4: Stattrib.txt File for Attributing Runoff Stations

A new directory called coverage is created on the UNIX system and the two input files, hydrost.dat and stattrib.txt, are copied to the directory. From the UNIX command prompt, type:

\$arc

to begin an Arc/Info session. From the arc prompt, type:

Arc: generate hydrost1

This command starts up a dialog for creating the point coverage and results in the generate command prompt. The command:

Generate: input hydrost.dat

specifies the source of the coordinates for building the points. Specify the type of feature to be created with:

Generate: points

The generation process begins to run. When this is complete, exit generate by typing:

Generate: quit

Individual points have now been generated but they have not been linked together into a single coverage. From the arc prompt, combine the generated points into a single, topologically referenced coverage by typing:

Arc: build hydrost1 points

Add the x and y coordinates to the feature attribute tables of the coverage by typing:

Arc: addxy hydrost1

The point coverage that has just been created is in geographic coordinates. It must now be projected into a flat map projection. As mentioned earlier, the coordinate system used in the Souss basin is the Lambert Conic projection. The parameters for this projection are stored in a text file called **sosprj** as shown in the Table 3.1. To project the generated coverage, hydrost1, to a new coverage, hydrost, using the projection file, sosprj, type:

Arc: project cover hydrost1 hydrost sosprj

from the arc command prompt. This creates a projected point coverage ready for use in the surface water model. Exit Arc/Info by typing:

Arc: quit

The creation of additional attributes can be done in Arc/Info or ArcView. However with a large number of attributes to be added, the use of a spreadsheet was found to be quicker. The addition of the attributes is done by opening the feature attribute table of the runoff station coverage, **hydrost.dbf**, in an Excel spreadsheet and along with the table containing the attribute data, **stattrib.txt**. The key fields, '**hydrost_id**' for **hydrost.dbf** and '**number**' for **stattrib.txt**, are compared to ensure that the records in the two tables are in the same order. The attributes in the coverage **hydrost** are updated by copying the desired columns from **stattrib.txt** and pasting them in **hydrost.dbf**.

After copying the attributes, the updated spreadsheet of **hydrost.dbf** is saved in Dbase(III) format to ensure that it can subsequently be read in **ArcView**. The option to save a spreadsheet to various formats is available under the **File/Save As...** menu in Excel. A similar process was followed in creating the point coverages of the rain gages, **rainst**, and the dams, **bigdams**. A list of point coverages developed, their key fields and other attribute fields is provided in Table 3.5.

<u>COVERAGE</u>	<u>KEY FIELDS</u>	<u>ATTRIBUTES</u>	<u>DESCRIPTION</u>
HYDROST	HYDROST_ID	HYDROST_ID	STATION NUMBER
	IRE	IRE	STATION ID (STRING)
		HYDROST_	COVERAGE ID
		STNAME	STATION NAME
		X	NORTHING
		Y	EASTING
RAINST	RAINID	RAINID	STATION ID (STRING)
		RAINST_ID	STATION NUMBER
		RAINST_	COVERAGE ID
		IRE	STATION NUMBER
		X	NORTHING
		Y	EASTING
BIGDAMS		Z	ELEVATION
	DAMID	DAMID	DAM NUMBER
		DAMS_	COVERAGE ID
		DAMS_ID	DAM NUMBER
		NAME	DAM NAME
		FIRSTYR	YEAR FIRST OPERATED
		FIRSTMTH	MONTH FIRST OPERATED

Table 3.5: Attributes of Souss Basin Point Coverages

3.3 PROCESSING RAINFALL DATA

The availability of good rainfall data is essential to the success of any surface water modeling effort. It is the primary input for the simulation model and consequently it greatly influences the quality of results. Geographic features such as mountains can result in unexpected rainfall patterns. Such patterns can only be represented in the model if a dense network of gauges is available. Hence data from all rain gauges located both within and close to the river basin were used. This enables spatial variations in rainfall patterns to be captured and represented in the model. Data from each gauge were used even if the period of record for the gauge covered only a portion of the study period. Important scenarios could otherwise have been excluded if only gauges covering the entire study period were used.

The rainfall data have to undergo a number of processing steps before being used in the simulation model. The first processing step involves transforming the data stored in two or three dimensional tables into single column time series. The traditional approach to gathering daily rainfall data involves recording measurements made at a remote station and transmitting this information to a central data bank, periodically. Consequently, data for each station are usually

stored in a separate file. These files are often two dimensional in nature with the month of the year as columns and the days as rows. Where paper filing systems are used, each year of data are often stored on a single sheet of paper resulting in a three dimensional file structure as illustrated in Figure 3.3.

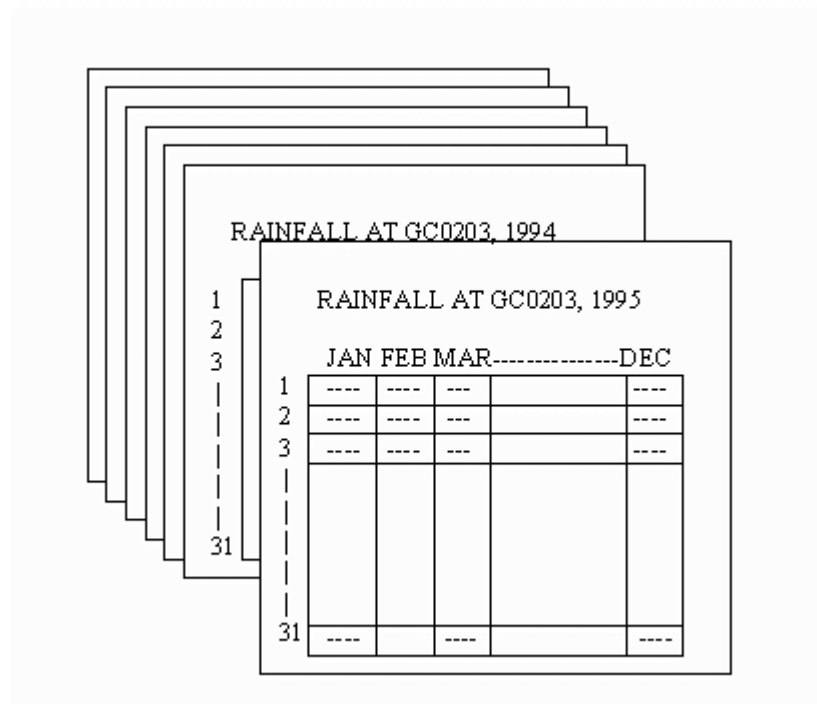


Fig. 3.3: A Typical Three Dimensional Data Structure

The object oriented programming approach used in the model requires that all data be stored as time series. This formatting problem can be solved in a number of ways. Cutting and pasting in a spreadsheet is an easy and interactive means of doing this conversion. However, with the 60 year period of record adopted in this study, the use of a spreadsheet was found to be too tedious to be

practical. A computer program developed in the programming language C, was used for this conversion. The program, called **sosprain.c**, was developed by Zichuan Ye of the Center for Research in Water Resources. This program is format dependent and was developed to perform the conversion of the rainfall data as acquired from the DGH in Morocco. The program is stored in the programs directory of the **Souss Basin Model** CD-ROM.

The next processing step involves interpolating for missing records. This proved to be the most tedious and time consuming task in developing the Souss basin model. The rainfall data used in the study were supplied by the Direction Generale de L'Hydraulique (DGH), the agency responsible for water resource development in Morocco. Daily rainfall records from 48 gauging stations, shown in Figure 3.4, were made available to the study. The data were collected over a period of 60 years from 1935 to 1994.

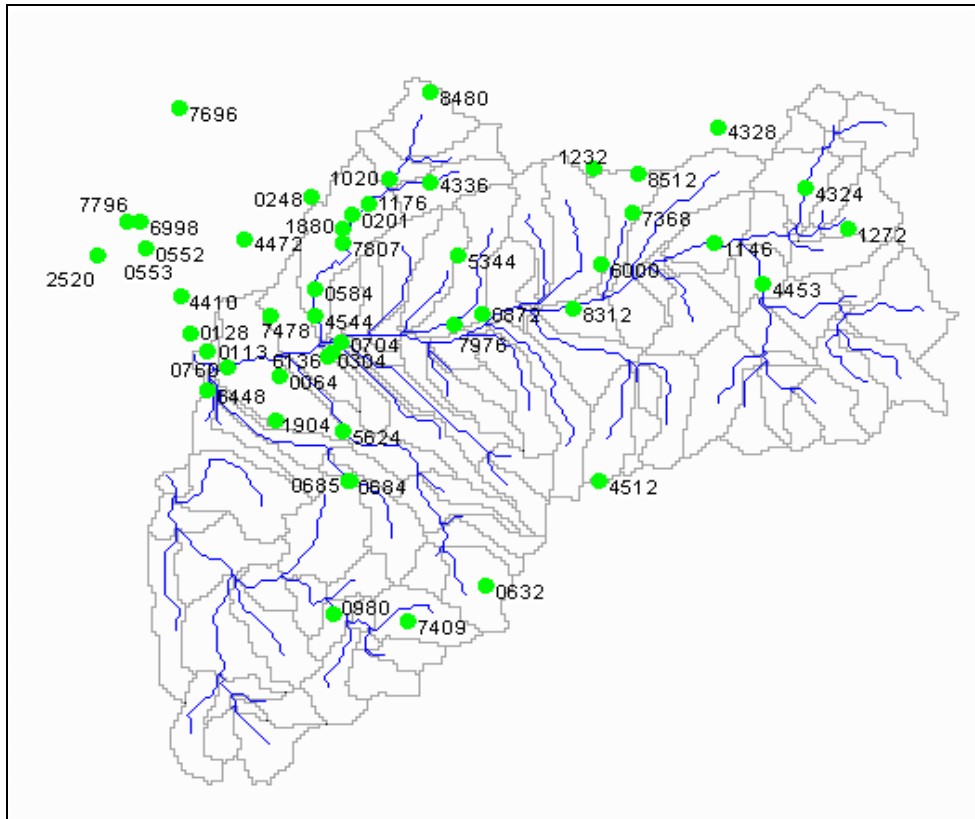


Fig. 3.4: Souss Basin Rain Gauging Stations

The rainfall data contained significant instances of missing records. In addition, the gauging stations did not have the same period of record. The programming approach used in **GH-Rivers** requires a rainfall record at each of the gauging stations during each time period. Hence, data from other stations are interpolated to fill in missing records and to generate records for periods when no data were collected. The **GH-Rivers** simulation model contains a script called **cmprain.ave** which generates data for missing records. This script runs through

the rainfall records until it locates a station with a negative (missing) record. It then uses the rain gage coverage, **Rainst**, to locate the closest gage with a non-negative rainfall reading for the same time period. It then assigns this rainfall reading to the missing record. Instructions for running this interpolation are included in section 4.1 of this report while the avenue script for the program is documented in appendix 4.

The interpolation can be run on a single file containing the whole period of record but since each year of data takes about an hour to process, this approach would have required a continuous 60 hour processing run. Such a run was not feasible because of practical considerations. Consequently, each year of data were extracted into a separate file with the station identification numbers as attributes and the days of the year as record numbers. The station identification numbers used are the same as those used by the DGH in Morocco. A sample daily rainfall file, rn80.dbf, is shown in Table 3.6.

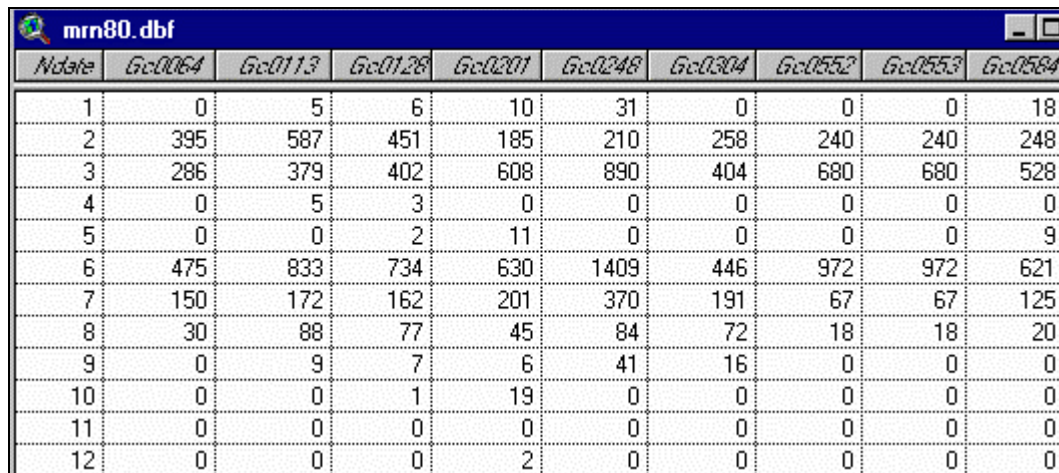
rn80.dbf							
Gc0984	Gc1020	Gc1146	Gc1176	Gc1232	Gc1272	Gc1880	Gc1904
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.44	0.00	0.00	1.22	0.55	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.60	0.00	0.00	0.68	0.02	0.00
0.00	0.00	14.83	0.00	26.00	13.07	2.29	0.00
0.00	6.00	10.31	0.00	0.00	10.99	2.69	0.00
0.00	0.00	4.83	0.00	7.00	5.10	0.15	0.00
0.00	0.00	1.16	0.00	8.00	0.83	0.05	0.00
0.00	0.00	0.58	0.00	4.00	0.41	0.03	0.00
0.00	0.00	5.01	0.00	0.00	5.74	0.12	0.00
0.00	0.00	1.43	0.00	0.00	1.64	0.04	0.00
0.00	0.00	12.01	0.00	45.00	10.95	0.44	0.00
0.00	0.00	2.24	0.00	0.00	2.09	0.02	0.00
0.00	0.00	4.90	0.00	0.00	5.63	3.91	0.00
0.00	0.00	1.88	0.00	13.00	1.35	0.09	0.00
0.00	0.00	1.61	0.00	11.00	1.15	0.08	0.00
0.00	0.00	4.41	0.00	0.00	5.05	0.11	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 3.6: Sample Daily Rainfall Data File

The file names of these annual rainfall files include a prefix, 'rn', to indicate the contents and two additional digits to indicate the year of record. For example, the rainfall data for all the stations in the basin for the year 1985 are stored in a file named 'rn85.dbf'. This resulted in 60 files (one for each year) containing 365 or 366 records with 48 fields each.

In addition to the interpolation, the data were also converted from daily to monthly records by summing up all the records at each station for each month. A

program called **Dmrain.ave** was developed to perform this conversion and write the results to a new file. The script for this program is documented in appendix 2 of this report. The naming convention adopted for the monthly data is the same as that used for the daily data except that the prefix ‘mrn’ is used instead of the previous ‘rn’. For example, the monthly data for 1985 are stored in a file named ‘mrn85.dbf’ as shown in Table 3.7. Consequently, there are 60 such tables containing monthly rainfall data from each of the 48 rain gauging stations in the Souss basin.



<i>Ndate</i>	<i>Gc0064</i>	<i>Gc0113</i>	<i>Gc0128</i>	<i>Gc0201</i>	<i>Gc0248</i>	<i>Gc0304</i>	<i>Gc0552</i>	<i>Gc0553</i>	<i>Gc0584</i>
1	0	5	6	10	31	0	0	0	18
2	395	587	451	185	210	258	240	240	248
3	286	379	402	608	890	404	680	680	528
4	0	5	3	0	0	0	0	0	0
5	0	0	2	11	0	0	0	0	9
6	475	833	734	630	1409	446	972	972	621
7	150	172	162	201	370	191	67	67	125
8	30	88	77	45	84	72	18	18	20
9	0	9	7	6	41	16	0	0	0
10	0	0	1	19	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	2	0	0	0	0	0

Table 3.7: Sample Monthly Rainfall Data File

The conversion from daily to monthly data was done to generate input data for the monthly model. The modeling of the Souss basin was initially performed at a daily time scale. This was subsequently changed to a monthly time scale because of difficulties encountered during the model calibration phase. These difficulties are discussed in section 6.1 of this report.

3.4 PROCESSING RUNOFF DATA

Spatially distributed parameters determine basin response in GH-Rivers. An optimization routine built into the surface water model determines the most suitable values of these parameters by calibrating flows generated by the model with the observed runoff. Hence runoff data collected at several runoff stations along the river network are required for calibrating the model. A database of runoff for selected stations around the world is also maintained by the Global Runoff Data Center in Koblenz, Germany. However, this database is not very extensive. For the Souss basin study, daily runoff records from six runoff stations were obtained from DGH in Morocco. The data files for the six runoff stations are listed in Table 3.8, and their locations are shown in Figure 3.5.

<u>Station Id</u>	<u>Station Name</u>	<u>Runoff Data File</u>
0203	Aoulouz	gc0203.dbf
0594	Amsoul	gc0594.dbf
0597	Aguenza	gc0597.dbf
0642	Immerguen	gc0642.dbf
0858	Ibergnaten	gc0858.dbf
4340	Ait Melloul	gc4340.dbf

Table 3.8: Souss Basin Runoff Data Files

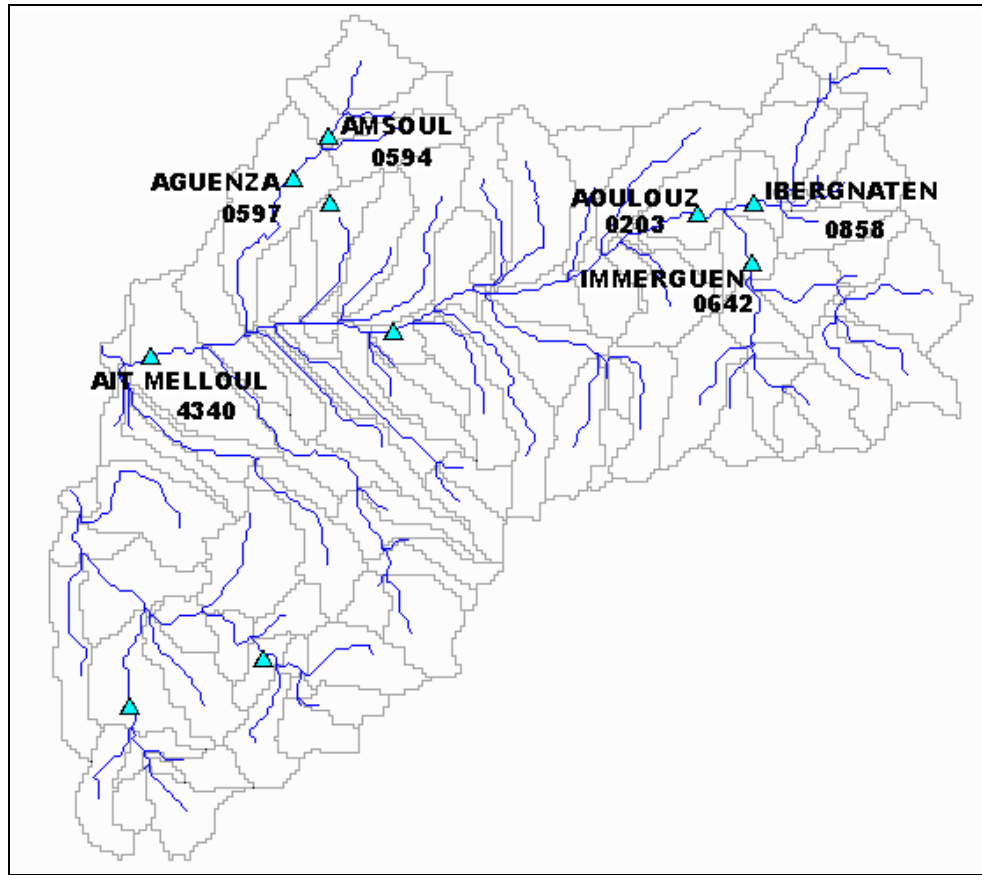


Fig. 3.5: Souss Basin Runoff Stations

Each of the stations had a different period of record although five out of the six stations contained records for the period 1979 to 1994. The sixth station located at Aoulouz has the longest period of record with daily values from 1954 to 1983. Figure 3.6 provides a comparison of the period of record of the six runoff stations.

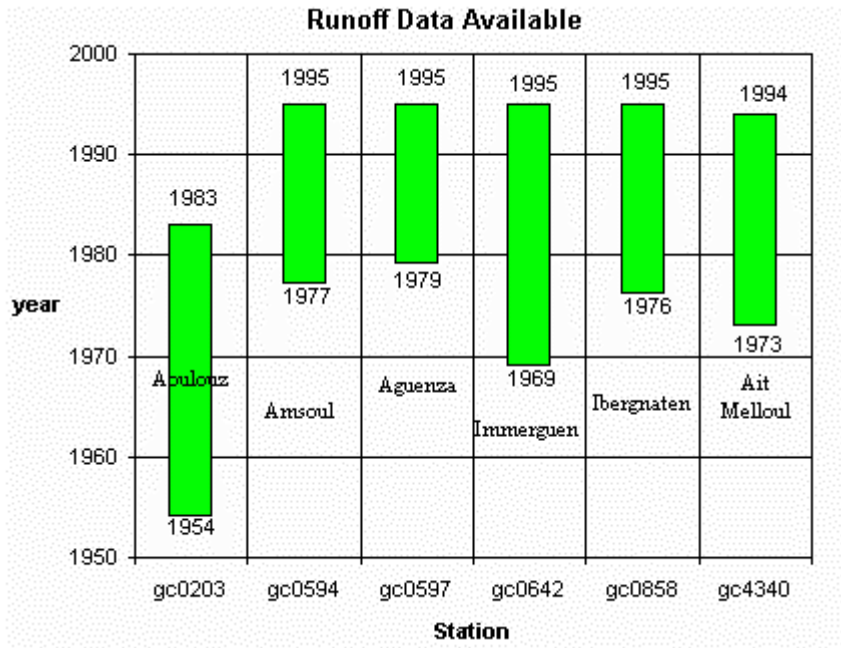
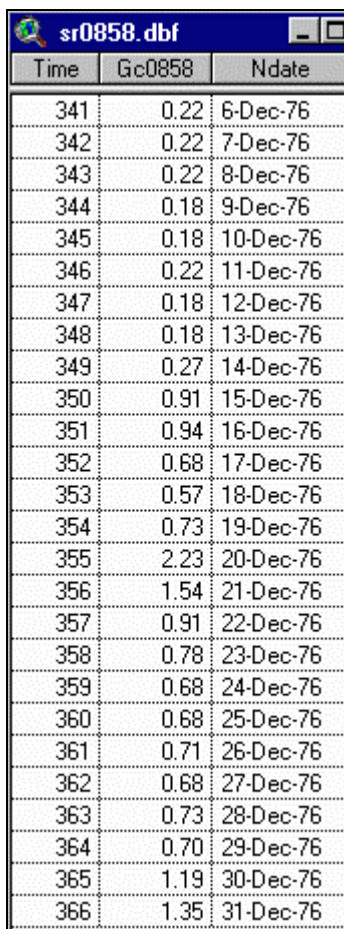


Fig. 3.6: Graph of the Runoff Stations Period of Record

The runoff data were stored in two dimensional tables with the station identification number (assigned by the DGH) as the file name. For example, the data for runoff station number '0203' were stored under the file name, 'gc0203.dbf'. Prior to using the data in the model, it had to be converted to a one dimensional time series. The C program, **sosprain.c**, used in converting the rainfall data was used for this conversion. (The program is stored under the **Programs** directory of the **Souss Basin Model** CD-ROM). The converted data were stored under file names 'sr0203.dbf' where 'sr' stands for surface runoff and the other four digits of the file name are the station identification number. The fields in this table include a time field ('Time') which acts as a key field for

keeping track of the records in the table, a date field ('Ndate') and a runoff field ('Gc0858' or the corresponding station identification number). The time field is indexed starting with 1 at the beginning of each year. A portion of the file is given in Table 3.9.



Time	Gc0858	Ndate
341	0.22	6-Dec-76
342	0.22	7-Dec-76
343	0.22	8-Dec-76
344	0.18	9-Dec-76
345	0.18	10-Dec-76
346	0.22	11-Dec-76
347	0.18	12-Dec-76
348	0.18	13-Dec-76
349	0.27	14-Dec-76
350	0.91	15-Dec-76
351	0.94	16-Dec-76
352	0.68	17-Dec-76
353	0.57	18-Dec-76
354	0.73	19-Dec-76
355	2.23	20-Dec-76
356	1.54	21-Dec-76
357	0.91	22-Dec-76
358	0.78	23-Dec-76
359	0.68	24-Dec-76
360	0.68	25-Dec-76
361	0.71	26-Dec-76
362	0.68	27-Dec-76
363	0.73	28-Dec-76
364	0.70	29-Dec-76
365	1.19	30-Dec-76
366	1.35	31-Dec-76

Table 3.9: One Dimensional Runoff Data File

As with the rainfall, one dimensional time series were generated for both daily and monthly runoff data. There are however two key differences in the format of the runoff time series. While rainfall values presented are daily or monthly totals, runoff values are presented in terms of mean flow rates. This difference becomes significant when converting daily runoff values into monthly values. An Avenue program called **Dmrunoff.ave** was developed for this conversion. The text of this program is documented in appendix 3 of this report. Daily runoff values are summed up and divided by the number of contributing days to form mean monthly values. Missing records do not have to be estimated prior to this conversion since this will not lead to an improved estimate of mean monthly flow rate. Instead, the runoff rate is averaged over the actual number of days in the month for which data were available. The second difference is that data for each runoff station were stored in a separate file. This is because in its current state, the optimization model does not automatically establish the linkage between runoff data and the stations in the runoff coverage. This linkage is established manually prior to each run of the optimization model. Consequently, no advantage is gained by providing a single table with all the runoff stations as fields. Such a table would require the establishment of a uniform period of record for all the runoff stations. Runoff data would hence have to be generated for periods when no data were collected. The establishment of a one to one

relationship between all fields in the input tables is a requirement of the object oriented programming approach used in developing the model.

A file naming convention that provides information about the contents of each data file was adopted. This convention includes information about the year of record, the station number and the period for which the runoff value is a representative mean. For example, daily runoff data for 1983 at station number gc0203 are stored under the filename, 'd83s0203.dbf'. The corresponding monthly data are differentiated from the daily data by replacing the first letter of the filename, 'd', with an 'm'. Thus, the equivalent monthly data for the same station and year are stored under the filename, 'm83s0203.dbf'.

Missing runoff values posed a problem in calibrating the daily simulation model. Methods of estimating missing runoff records do not appear to be well documented in the literature. Those methods that are documented tend to involve the use of spatial relationships. Such methods cannot be used for this study because the runoff data are intended to provide a means of calibrating spatially distributed parameters and determining spatial relationships. Only those methods which exploit temporal relationships without taking spatial variation into account can be used in this model.

One of the approaches considered in estimating missing runoff records is the averaging of the flow from days preceding and subsequent to the missing record. There are two main limitations to this approach. Visual inspection of the runoff data from the Souss basin indicates that such an estimate can be misleading during periods of high rainfall. Significant variation in daily flow rates are observed at stations in the basin during such periods. However, this would not have posed a problem since missing runoff records tended to occur during low flow periods. The second limitation of this approach is that estimates of individual missing records cannot be made when a string of consecutive records are missing. The missing daily flow rates would have to be estimated by assuming a constant flow rate over the entire period of missing records. Again, it was determined by visual inspection that strings of missing records tended to occur after strings of readings with little variation between them. During such periods, there was little variation between the mean monthly flows and the observed daily flows. Consequently, the mean monthly flows could be substituted for the missing daily records.

An alternative approach to solving the problem of missing data is to leave periods with missing data out of the calibration of the daily model. Monthly runoff data are unaffected by such missing daily records, since monthly values can be determined from available daily values. However, with the scarcity of good runoff

data, it may not always be possible to leave out periods with missing data especially when such periods include major hydrologic events such as floods or droughts. For the Souss basin simulation, only periods with no missing records were used in the calibration. No interpolation has been performed to estimate missing records in the daily records.

3.5 PREPARING RESERVOIR DATA

For planning efforts to be effective, there must be a means of controlling the availability of water, other than the natural processes of rainfall and runoff. Dams and their accompanying reservoirs represent the most significant means of controlling the quantity of water in storage. They are also used in regulating flow in the river channel downstream of the dam. Hence, it is important to keep track of the conditions in the reservoirs.

The Souss basin contains three major reservoirs located at Aoulouz, Abdel Moumen and Dhkila, as shown in Figure 3.7. Direction Generale de L'Hydraulique (DGH), Morocco, provided detailed operation data covering the entire period of operation of each of these reservoirs. They also supplied reservoir regulation data used in defining the storage-elevation and storage-area curves for each of the three reservoirs. These reservoirs were assigned identification numbers

1, 2 and 3, respectively. Table 3.10 summarizes the main characteristics of the three reservoirs. The annual inflow presented are averaged over the entire period of operation of each reservoir. The maximum storage and surface area of each reservoir is also provided.

NAME	DAMID	FIRST YEAR	STORAGE (m ³)	AREA (m ²)	ANNUAL INFLOW (m ³)
Aoulouz	1	1991	351,458,300	13,000,000	77,247,000
Abdel Moumen	2	1982	244,508,200	7,921,700	62,564,000
Dhkila	3	1986	1,525,100	214,600	73,611,000

Table 3.10: Characteristics of the Souss Basin Reservoirs

The dam at Aoulouz is mainly used for regulating flow in the portion of the river flowing over the aquifer, downstream of the dam. Over 70% of all inflow to this reservoir is subsequently released for aquifer recharge. Ground water is an important source of water in the basin and a number of extraction wells have been dug to provide water for irrigation. By regulating releases at this dam, the amount of water that percolates into the underlying aquifer is maximized. Some releases are also made, from Aoulouz, for irrigation purposes. However, these releases only account for about 1% of annual inflow. The rest of the inflow is either released to

satisfy downstream flow requirements (17%) or lost through evaporation (4%) and leakage (3%).

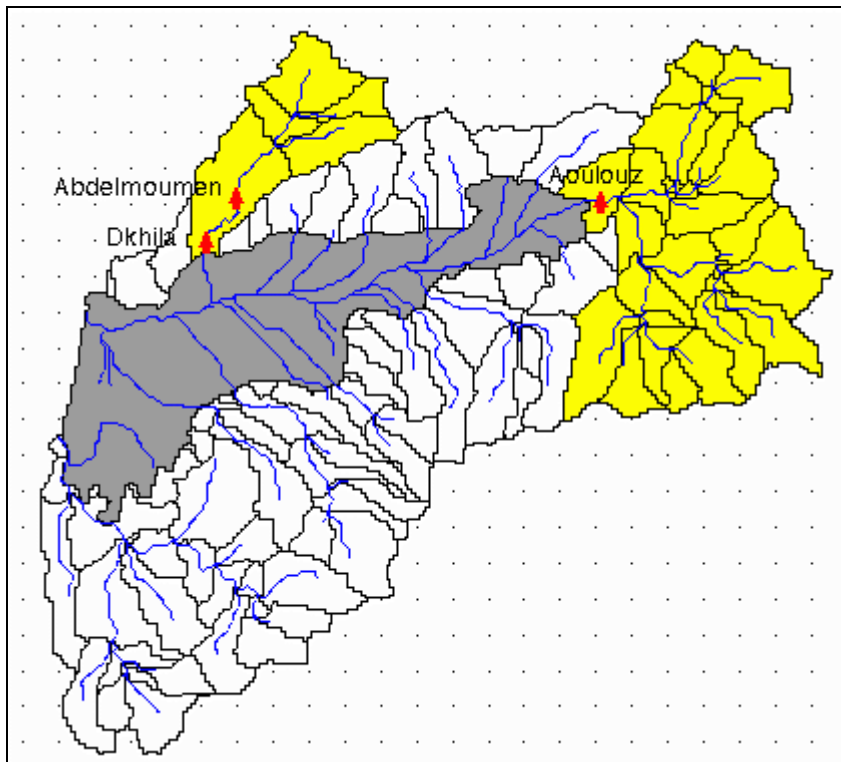


Fig. 3.7: Souss Basin Dams and Aquifer

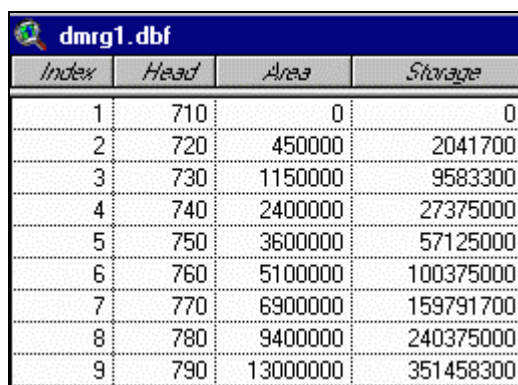
The dam at Dkhila is used for irrigation and water supply. Over 50% of annual inflow is diverted for irrigation. Another 13% is withdrawn to supply water for domestic and industrial purposes in the resort city of Agadir. Releases for these uses are made through a canal located at the side of the reservoir. Located upstream of Dkhila, the dam at Abdel Moumen helps to minimize releases by

storing excess flow during high flow periods. Releases are subsequently made from Abdel Moumen to Dhkila during low flow periods. About 34 million cubic meters, 50% of annual inflows, are released from Abdel Moumen to Dhkila every year.

The important characteristics for water resource planning are the volume of water in storage, the surface area and the water surface elevation. The relationship between the volume, area and elevation for a given reservoir is determined at the time of construction of the reservoir. This relationship is presented in the form of a series of equations or curves. It may also be presented in the form of reservoir regulation tables containing corresponding values of storage, surface area and water surface elevation taken at regular intervals. Reservoir regulation tables are optional inputs for GH-Rivers but they must conform to a specified data format if used in the model. The fields in these tables include a key field ('Index'), a water surface elevation field ('Head') in meters, a surface area field ('Area') in meters squared and a storage field ('Storage') in cubic meters, as shown in Table 3.11.

For the Souss basin study, reservoir regulation data were obtained in terms of values recorded at regulation intervals. Data for each reservoir in the river basin are stored in a separate file. A naming convention was adopted to enable the program to form an association between the data files and the reservoir objects

during the simulation. The data were stored under filename such as dmrg2.dbf where the first four characters indicate that the file contains dam regulation data. The remaining digits of the base name are the identification number defined in the reservoir object of the model; 1 for Aoulouz, 2 for Abdel Moumen and 3 for Dhkila. If the regulation data are presented as either equations or curves, then they must be converted to tabular format. Storage elevation curves can be plotted from equations and reservoir regulation tables extracted from these curves. The extraction of these tables can be done by reading storage and surface area values off the storage-elevation and surface area-elevation curves at regular intervals of elevation. The units and field names described in the regulation tables below should be maintained in all reservoir regulation tables used in the model. The regulation files for Aoulouz, Abdel Moumen and Dkhila are presented in Tables 3.11, 3.12 and 3.13, respectively.



<i>Index</i>	<i>Head</i>	<i>Area</i>	<i>Storage</i>
1	710	0	0
2	720	450000	2041700
3	730	1150000	9583300
4	740	2400000	27375000
5	750	3600000	57125000
6	760	5100000	100375000
7	770	6900000	159791700
8	780	9400000	240375000
9	790	13000000	351458300

Table 3.11: Dam Regulation File for Aoulouz

dmrg2.dbf			
<i>Index</i>	<i>Head</i>	<i>Storage</i>	<i>Area</i>
1	575.0	0	0
2	577.5	4600	3700
3	580.0	60500	48800
4	582.5	300400	140500
5	585.0	740700	216500
6	587.5	1415000	321400
7	590.0	2340600	417100
8	592.5	3466900	500900
9	595.0	4959100	686500
10	597.5	6845500	833600
11	600.0	9200100	1046700
12	602.5	12048100	1239300
13	605.0	15448900	1477900
14	607.5	19402900	1695500
15	610.0	23981400	1974500
16	612.5	29334500	2296800
17	615.0	35388400	2551800
18	617.5	42122800	2840000
19	620.0	49618500	3153900
20	622.5	57854600	3451500
21	625.0	66985400	3848500
22	627.5	77057800	4217300
23	630.0	88128500	4633500
24	632.5	100202900	5015200
25	635.0	113133100	5331900
26	637.5	126881200	5666100
27	640.0	141462700	5997800
28	642.5	156862200	6321600
29	645.0	173088300	6644300
30	647.5	189989900	6877000
31	650.0	207437200	7109700
32	652.5	225718900	7515700
33	655.0	244508200	7921700

Table 3.12: Reservoir Regulation File for Abdel Moumen

dmrg3.dbf			
<i>Index</i>	<i>Head</i>	<i>Area</i>	<i>Storage</i>
1	215.00	3400	0
2	216.00	6300	1500
3	217.00	8900	9100
4	218.00	11000	19000
5	219.00	14000	31200
6	220.00	20400	48500
7	221.00	25400	71500
8	222.00	29900	99100
9	223.00	34800	131000
10	224.00	45200	170900
11	225.00	56100	221700
12	226.00	66000	282400
13	227.00	79800	354700
14	228.00	100700	445200
15	229.00	118300	555200
16	230.00	130400	679800
17	230.25	132900	712700
18	231.00	143200	816000
19	232.00	163600	969800
20	233.00	179500	1141300
21	233.25	183500	1186700
22	234.00	196100	1329000
23	235.00	214600	1525100

Table 3.13: Reservoir Regulation File for Dhkila

When using the model to perform a simulation, surface area and water surface elevation values are determined from the storage at the end of each time step. The values are determined by interpolation from the reservoir regulation tables. The GH-Rivers simulation model will run without the regulation files by using an in-built synthetic equation to estimate the surface area for a given volume in storage. However, regulation tables should be used whenever possible as they

provide a much more accurate estimate of surface area. Also, water surface elevations are not computed when the synthetic equation is used. The synthetic equation is as follows:

$$(A^t / A_{\max}) = (S^{t-1} / S_{\max})^{0.72}$$

where

A^t is the surface area of the reservoir at the beginning of period t

A_{\max} is the maximum surface area of the reservoir

S^{t-1} is the volume of water in storage at the end of period $(t-1)$

S_{\max} is the maximum reservoir storage

Reservoir operation data are also required to give a clear picture of the historical operation patterns. The operation data for the three reservoirs in the Souss basin were obtained as paper documents, written in French. The documents were translated into English before being transcribed into electronic format. A list of the fields in the original reservoir operation tables and how they were translated from French to English is presented in table 3.14.

<u>Field Name in French</u>	<u>Translated Field Name</u>
Annee	Year
Cote	Elevation
Volume	Storage
Surface	Surface Area
Variation Reserve	Change in Storage
Evaporations	Evaporation Losses
Restitution Non-Controlees	Uncontrolled Releases
Restitution Dhkila	Releases to Dhkila
Restitution Fuites	Leakage
Restitution Vidanges	Minimum Flow Releases
Restitution Deverses	Spillage
Restitution Total	Net release
Apports	Inflow

Table 3.14: Translation of Reservoir Operation Fields

There is a separate document for each of the three reservoirs. Each document includes a summarized table of annual inflows, evaporation losses, releases and net change in storage for each water year during which the reservoir was in operation. The table also contains the reservoir elevation, volume and surface area recorded on the first day of each water year (August 1st). A

supplementary summary table contains the total, mean, maximum, minimum and coefficient of variation for each of the parameters listed above, determined over the entire period of record. In addition to these annual values, monthly records of the inflow, releases to various uses, leakage, evaporation and spill are presented in separate tables each with an accompanying summary table similar to the one described for the annual table. These are stored as two dimensional tables as illustrated by the sample monthly release Table 3.15.

Year	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Total
81-82						0	0	0	0	0.38	0	0	
82-83	0	0	0	0	0	0	0	0	0	0	0	0	2E-04
83-84	0	0	0	0	0	0.79	0.63	0.01	0	0	0.63	1.13	3.196
84-85	0.01	0.01	0	0	0	0	1.18	1.96	0	0.04	1.56	3.49	8.347
85-86	0.09	0.8	0.76	0.54	0.85	0.73	0.63	0.8	3.37	1.56	3.25	1.98	15.35
86-87	1.04	1.34	0.93	1.85	2.21	0.86	2.58	4.4	1.81	2.37	2.63	3.88	25.89
87-88	4.07	2.75	0.76	0.53	0.61	1	0.51	4.05	4.14	3.86	4.17	4.62	31.08
88-89	4.57	2.15	0.17	4.43	5.33	3.19	5.69	3.46	4.9	5	4.45	5.04	48.36
89-90	5.04	3.36	1.72	0.21	0.77	4.54	3.09	3.26	5.04	4.85	4.91	5.26	42.06
90-91	4.73	4.06	3.8	2.18	3.63	2.96	2.38	4.65	5.78	5.41	5.16	5.67	50.4
91-92	6.08	5.01	4.47	2.41	5.96	5.74	8.21	6.78	7.04	6.68	6.3	6.78	71.45
92-93	7.27	5.41	5.94	5.12	5.16	5.14	7.26	8.69	5.85	5.67	6.02	6.01	73.54
93-94	6.49	5.05	0.68	2.9	4.45	5.21	5.46	5.76	5.28	4.96	6.04	5.76	58.04
94-95	5.64	5.16	4.08	4.43	3.32	2.81	1.13	0.69	1.1	1.21	3.18	4.35	37.09

Table 3.15: Uncontrolled Monthly Releases at Abdel Moumen

The data were transformed into one dimensional time series during the transcription process by typing all the records into a single column as shown in

Table 3.18. Even though the documents were developed by the same agency, differences existed in the definition of the reservoir outflows used in the regulation tables of the three dams. Hence, the outflows were reclassified into a number of uniform categories including leakage, evaporation, withdrawals and releases. The leakage and evaporation fields both result in water being lost from the river network, and they are common to all three reservoirs. These fields were consequently left unaltered in the redefined reservoir operation tables. Other reservoir outflows were reclassified as releases if they resulted in water reentering the river channel downstream of the dam, or withdrawals if they resulted in water being removed from the river network entirely. The resampled fields were stored as time series under the file names `aouldmop.dbf`, `abdmdmop.dbf` and `dkildmop.dbf` for the reservoirs at Aoulouz, Abdel Moumen and Dhkila, respectively. Table 3.16 shows the outflow fields in the operation tables of each reservoir and the equivalent field into which they were reclassified.

<u>Original Field</u>	<u>Outflow Type</u>	<u>New Field Name</u>
For Dam 1, Aoulouz		
Uncontrolled Releases	Release	Release
Aquifer Recharge	Release	Release
Irrigation in Aoulouz	Withdrawal	Withdraw
Leakage	Losses	Leakage
Evaporation	Losses	NetEvap
For Dam 2, Abdel Moumen		
Uncontrolled Releases	Release	Release
Release to Dhkila	Release	Release
Leakage	Losses	Leakage
Evaporation	Losses	NetEvap
Spillage	Release	Release
Minimum Flow Release	Release	Release
For Dam 3, Dhkila		
Evaporation	Losses	NetEvap
Leakage	Losses	Leakage
Irrigation	Withdrawal	Withdraw
Water Supply	Withdrawal	Withdraw
Minimum Flow Release	Release	Release
Spillage	Release	Release

Table 3.16: Original and Resampled Reservoir Operation Fields

Inflows ('Inflow') provided in reservoir operation data may include streamflow as well as estimates of direct rainfall and surface runoff, and

springflow from subsurface water. Leakage ('Leakage') values represent losses through infiltration at the base of the reservoir. Values of net evaporation ('NetEvap') are determined from field measurements of open water surface evaporation and presented in meters of water. Withdrawals ('Withdraw') refer to flows that are taken from the reservoir to meet demands. These flows do not return to the river channel. Releases ('Release') on the other hand refer to outflows from the reservoir that go directly back into the river channel. Releases subsequently become available for use downstream of the reservoir. A key field ('Time') is provided to keep track of the simulation time step while a date field ('Date') keeps track of the period of record. Additional fields were also defined to store the results of simulation runs. These include the volume of water lost through evaporation ('EvplLoss'), reservoir storage ('Storage'), surface area ('SArea') and head ('Head') at the end of the time step, inflow generated by the model ('Cinflow') and spill generated whenever the volume of water in storage exceeds the reservoir capacity. Though the demand factor field ('Dmdfact') was not used in this model, it was retained in the operation tables because it was included in the original simulation routine. A complete listing and description of the fields in the operation tables is provided in Table 3.17.

Parameter	Description	Units
Time	The time step being simulated, starting with 1	-
Date	Time period extracted from historical operation data	-
Dmdfact	Demand Factor: Fraction of demand to be met	-
Netevap	Net evaporation from open water surface	m
Head	Water surface elevation above mean sea level	m
Sarea	Water surface area at the end of the time period	m ²
Withdraw	Water removed from reservoir and not returned to river	m ³
Leakage	Water seeping through dam and reservoir base	m ³
Evploss	Volume of water lost due to evaporation	m ³
Spill	Uncontrolled releases due to excess in storage	m ³
Storage	Volume of water in storage at the end of time period	m ³
Release	Water released from reservoir into river channel	m ³
Cinflow	Inflow computed by surface water simulation model	m ³
Inflow	Inflow stated in historical reservoir operation tables	m ³

Table 3.17: Parameter Description for Dam Operation Table

A sample dam operation table for the reservoir at Abdel Moumen is presented in Table 3.18. Each record in this table represents a month of data. The surface area (Sarea), volume of water lost through evaporation (Evploss), reservoir storage (Storage), water surface elevation (Head) and generated inflow (Cinflow) fields do not yet contain any data. These fields are updated by the reservoir simulator when a simulation is performed.

TIME	INFLOW	DMDFACT	RELEASE	WITHDRAW	NETEVAP	LEAKAGE
1	4.45	0.000	0	0	0.095	0.001
2	0.599	0.000	0	0	0.133	0.001
3	5.234	0.000	0	0	0.203	0.001
4	8.722	0.000	0	0	0.262	0.002
5	4.474	0.000	0.379	0	0.422	0.002
6	0.722	0.000	0	0	0.351	0.002
7	0.051	0.000	0	0	0.575	0.003
8	0	0.000	0.091	0	0.553	0.003
9	0	0.000	0.259	0	0.374	0.003
10	0	0.000	0.1122	0	0.349	0.003
11	1.586	0.000	0	0	0.254	0.003
12	0	0.000	0.045	0	0.215	0.003
13	0	0.000	0.175	0	0.248	0.002
14	0	0.000	0.171	0	0.196	0.003
15	0.052	0.000	0	0	0.377	0.003

SPILL	DATE	SAREA	EVLOSS	STORAGE	HEAD	CINFLOW
0	Jan-82	0	0	0	0	0
0	Feb-82	0	0	0	0	0
0	Mar-82	0	0	0	0	0
0	Apr-82	0	0	0	0	0
0	May-82	0	0	0	0	0
0	Jun-82	0	0	0	0	0
0	Jul-82	0	0	0	0	0
0	Aug-82	0	0	0	0	0
0.0005	Sep-82	0	0	0	0	0
0.0002	Oct-82	0	0	0	0	0
0	Nov-82	0	0	0	0	0
0	Dec-82	0	0	0	0	0
0	Jan-83	0	0	0	0	0
0	Feb-83	0	0	0	0	0
0	Mar-83	0	0	0	0	0

Table 3.18: Reservoir Operation Table for Abdel Moumen

Prior to each run of the simulation model, records corresponding to the simulation period are extracted into separate tables for each reservoir in the basin. The names of these tables contain the dam identification numbers ('Damid') which

enable the model to establish the linkage between the operation tables and the dams coverage. The first three digits of the file name indicate that the table contains dam operation data and the remaining digits of the base name are the dam identification number ('damid'). The operation data for the reservoir at Aoulouz are extracted into the table, 'dam1.dbf' data while the data for Abdel Moumen and Dhkila are extracted into 'dam2.dbf' and 'dam3.dbf', respectively.

Chapter 4: Modeling the Souss Basin

Having assembled the input data and coverages, the extent of the basin to be modeled must be reviewed to ensure that it reflects the quantity and distribution of data as well as the study objectives. Isolated river segments and parts of the streams that cross the river basin boundary without first joining the main river network must be removed. This is necessary to ensure that the river network forms a closed system which can be calibrated. The editing of coverage can be performed in ArcView by simply selecting and deleting unwanted arcs. However, the Arc Edit module of Arc/Info software provides a more powerful and interactive edit environment. Instructions for editing coverages in Arc Edit were provided in section 3.2 of this report. The editing of the stream network for the Souss basin was not done as thoroughly as it should have been because the effects of such streams were not fully understood at the time of development of the model. The effects of this oversight on the overall simulation effort are discussed in section 6.1 of this report.

All the rainfall and runoff data available from the Souss basin were processed to ensure that the model produced could be used to simulate any period for which data were available. However, the simulation model can only be

developed and calibrated for a limited number of time steps. Both the daily and monthly models were constructed for a simulation period of one year in line with the setup of the rainfall and runoff data into individual years. This time period also allowed the reservoirs to be calibrated through a full cycle of the hydrologic conditions that occur during a calendar year.

4.1 DEVELOPING THE MODEL

The next phase of the simulation effort is to assemble the coverages and data created into a model. In addition to the data processing described in chapter 3, the coverages have to be preprocessed to convert them into a format usable by the GH-Rivers simulation program. There are 7 preprocessing steps which have to be completed prior to the simulation as shown in Figure 4.1. The first five steps realign the basin coverages and prepare them for use in the object oriented environment of the simulation model. The sixth and seventh steps create the output files and coverages required by the model. An alternative one-step, preprocessing function has been developed, however the use of this function precludes interaction between the user and program. The preprocessing continues even if error messages are displayed during the processing. Such errors can render the resulting coverages unusable. The functions performed by the seven preprocessing

steps are described below. Following these descriptions, instructions are provided on how to perform the preprocessing in the ArcView based model.

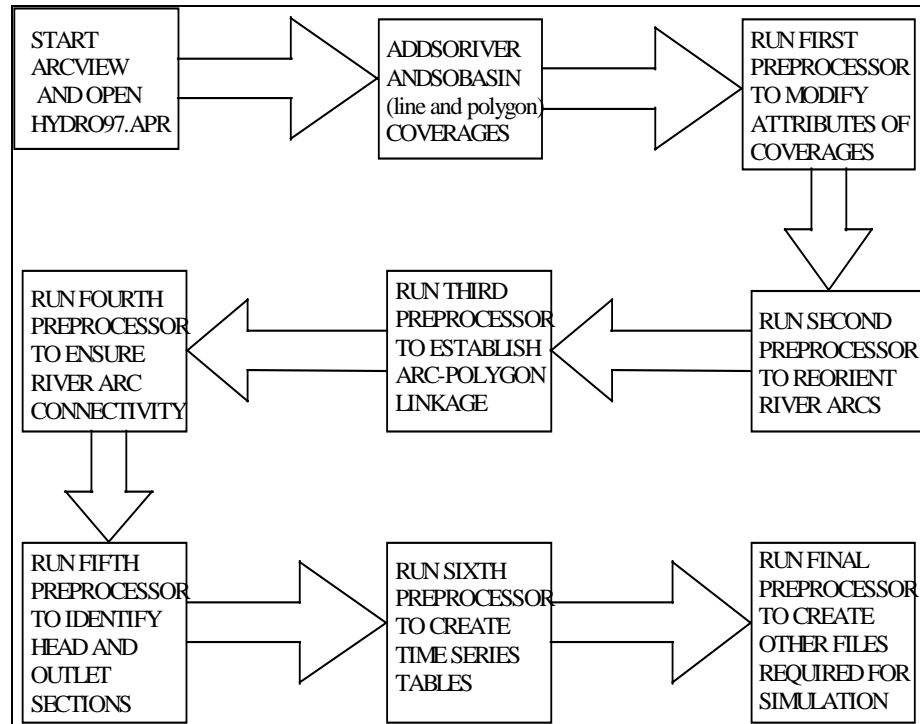


Fig. 4.1: Flow Diagram of Model Preprocessing


In the first preprocessing step, the user is prompted to identify the basin coverages to be used in the model. Attributes required for subsequent operations are added to the feature attribute tables of the coverages. In the second step, the arcs in the river network are reordered to ensure that water flows downstream throughout the network. River head and outlet sections are also identified and

labeled in the feature attribute table. The third, fourth and fifth steps are non-interactive. They establish a one-to-one relation between arcs in the river coverage and polygons in the watershed coverage. Such a relationship allows surplus water from the watershed to be directed to the appropriate river arc. These steps also ensure that there is continuity in the river network by establishing the arc topology. Nodes are redefined such that all river arcs are interconnected and pointing away from head sections towards outlet sections. This is necessary to ensure that flow in each arc in the river network can be passed on to the next downstream arc until it eventually reaches an outlet node.

The sixth step creates data files for storing the input rainfall and runoff time series data. The user is prompted for the initial and final time steps. The number of records in the tables is set to cover the specified time horizon. All subsequent simulations using this model must be performed with this number of records. The same number of records are added to the tables created in the seventh and final step. The tables created in this step are used for storing the results of simulation computations. The preprocessing may have to be repeated in a new workspace if a simulation is to be performed over a new time horizon to ensure that simulation tables contain the correct number of records. The modified preprocessing procedure described in section 4.4 can be used for this run to avoid reinitializing the values of model parameters.

To begin performing the preprocessing, a new workspace called **souss** is created and the ArcView project, **soussday.apr**, is copied to the workspace. The **soussday.apr** project file is used for the daily model while **soussmon.apr** is used for the monthly model. The project files for both of these models are stored in the respective **daily** and **monthly** directories under the **model** directory of the **Souss Basin Model** CD-ROM. This is necessary because the GH-Rivers currently does not contain an input box for specifying whether the model being run on a daily or monthly time scale. A couple of lines in the routing program must be modified to make this distinction.

The project files contain the surface water and reservoir planning extensions, SFlwModel and Dams. An extension is a suite of programs containing several stand-alone menus which can be created by users to customize the original ArcView interface. It is a new feature introduced in ArcView version 3.0 to provide users with more freedom in developing the software to suit their particular needs.

A new View is opened by double clicking on the View icon in the project window. The coverages of the Souss River basin are added to the ArcView project as themes by clicking on the  tool in the View Menu Bar and browsing through the directories to select the coverages. The basin coverages added to the new View include the line coverage of the stream network, **soriver**, the polygon coverage of the watersheds, **sobasin**, and the line coverage of the watershed boundaries, **sobasin**. The point coverages of the rain gages, **rainst**, the runoff stations, **hydrost** and the dam locations, **bigdams**, are also added to the project.

The one-step preprocessing procedure is run by selecting SGFPrep (pre-all) from under the SFlwModel in View Menu Bar as shown in Figure 4.2.



Fig. 4.2: Diagram of Surface Water Menu

A series of dialog boxes come up for the user to provide information for the preprocessing. The information requested by the dialog boxes and the inputs provided in developing the monthly model of the Souss basin are summarized in Table 4.1.


DIALOG NUMBER	INFORMATION REQUESTED	INPUT OPTIONS	INPUT PROVIDED
1	Pick a River Line Theme	Soriver , OK / Cancel	Soriver , Click OK
2	Pick a Basin Polygon Theme	Sobasin , OK / Cancel	Sobasin , Click OK
3	Pick a Basin Boundary Theme	Sobasin , OK / Cancel	Sobasin , Click OK
4	[A list of Ftab Modifications made]	OK	Click OK
5	Choose the feature to highlight	Both, Head, Outlet OK / Cancel	Both , Click OK
6	Highlight Exits ?	Yes / No	Yes
7	Highlight Exit ?	Yes / No	Yes
8	Save Control file as sfdbfs.ctl	(workspace directory) OK / Cancel	(workspace directory) , OK
9	[A list of elements in the control list]	OK	Click OK

10	Choose the feature to highlight	Both, Head, Outlet OK / Cancel	Click OK
11	Create time Series Tables	(Multiple Input List) OK / Cancel	Click OK
12	Directory to Write Files to	(workspace directory) OK / Cancel	(workspace Directory) , OK
13	Records to be Added to Time Series Tables	(Number of Records) OK / Cancel	12 , Click OK
14	[A list of files created]	OK	Click OK
15	Directory to Write Files to	(workspace directory) OK / Cancel	(workspace Directory) , OK
16	[A list of files created]	OK	Click OK
17	Keep Graphics	Yes / No	No

Table 4.1: Request and Inputs for the Model Preprocessing


With the setup of the model complete, the next step is to input rainfall data and perform a trial simulation to ensure that the model has setup correctly. The rainfall data used in the model must contain a record for every time period. If no records are made during a particular time period, records from adjacent stations must be interpolated to obtain estimates of such missing records. The simulation program contains a script, CMPRAIN.PRE, for performing such interpolations.

The script is included in Appendix 4. It assumes that missing records are stored as -1 or some other negative number in the rainfall time series. Rainfall recorded at the nearest station for the same time period is assigned to the missing record.

To run CMPRAIN.PRE, click on the run tool, , from the View Tool Bar and then click in the View containing the Souss basin coverages. A dialog box comes up with a default program script listed. To run the default program, select YES. If the script listed is not CMPRAIN.PRE, then click NO. An input screen comes up with a list of all the available scripts. Scroll down to select CMPRAIN.PRE and click OK. This initiates a dialog to specify the rain station theme (rainst) and the rainfall time series (e.g. mrn91.dbf). The program begins to interpolate for missing records, highlighting the edit station in the View as it goes. The results of interpolation are written back to the same time series table, mrn91.dbf in this example.

Each time a simulation is performed for a new period of record, the surplus computation program, CMPSURP.PRE, must be run to generate input for the surface water simulation. CMPSURP.PRE interpolates rainfall measured at rain gages to the centroid of the polygons in the subwatershed coverage, Sobasin, using an Inverse Distance Weighting scheme. It then computes the surplus as the flow that would be generated at the downstream node of the river arc in each polygon if

all the rainfall incident on the polygon were to become available as runoff. This surplus is computed in cubic meters per second as the incident rainfall in meters multiplied by the area of the subwatershed polygon in meters squared. This definition of surplus is used in the model because it was initially developed to use available moisture computed from soil-moisture balance computations as the source of inflow. Keeping this definition of surplus ensures that either the available moisture from the soil-moisture balance or total moisture from incident rainfall can be routed with the same programs since routing occurs after the generation of surplus.

Use the  tool to run CMPSURP.PRE. As previously, click on the run tool from the View Tool Bar and then click in the View containing the Souss basin coverages. This time the default program script listed is the last script run, in this case, CMPRAIN. Click NO and scroll down to select CMPSURP.PRE from the resulting list of scripts. A series of dialog boxes come up for the user to specify the inputs required by the program. The information requested by the dialog boxes and the inputs supplied are shown in Table 4.2.

DIALOG NUMBER	INFORMATION REQUESTED	INPUT OPTIONS	INPUT PROVIDED
1	Select a Rainfall Time Series Table	(A list of the available tables) OK / Cancel	(rainfall time series e.g. mrn87.dbf) , OK
2	Rain Station Theme	(A list of the point coverages) OK / Cancel	Rainst , OK
3	Enter Watershed Theme To Interpolate Rainfall	(A list of the polygon coverages) OK / Cancel	Sobasin.ply , OK
4	Enter Watershed Key Field e.g. Grid-code	(A list of attributes of the watershed theme) OK / Cancel	Grid_code , OK
5	Enter Rain Station Key field	(A list of attributes of the rain station theme) OK / Cancel	Rainid , OK

Table 4.2: Requests and Inputs for the Surplus Interpolation

The interpolation begins with a graphical display in the View indicating which stations have been computed as well as the station currently being computed. Figure 4.3 shows the graphical display in View at the end of the interpolation, with all the centroids of the watersheds identified.

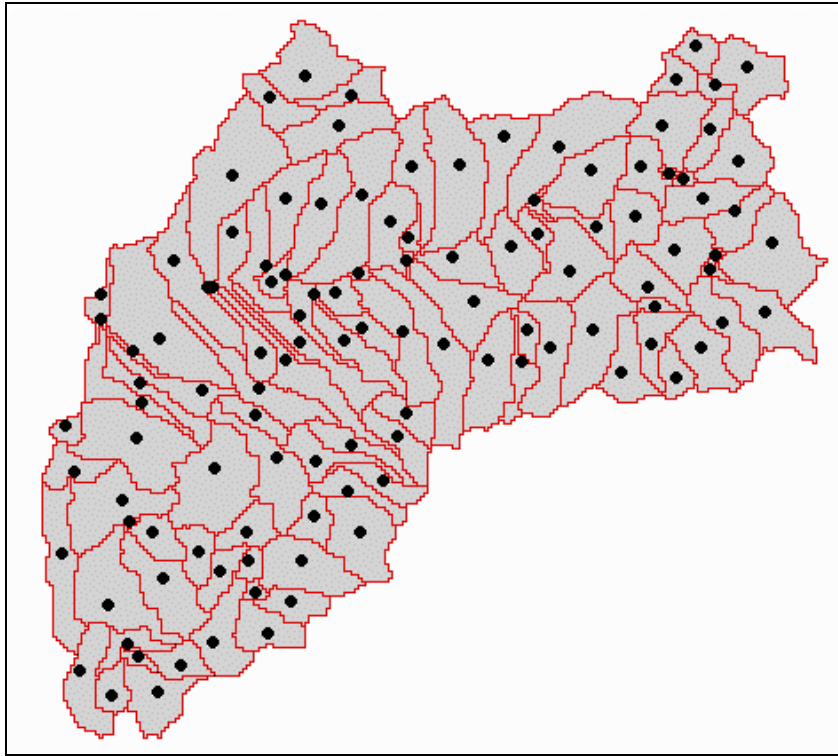


Fig. 4.3: Interpolation to Centroid of Watershed Polygons

When the computation is complete a window appears to inform the user that the computation is complete. Click OK to dismiss the window. The results of the computations are written to the surplus table, 'psurp.dbf'. For each watershed in the basin there is a corresponding field in psurp.dbf. The table is linked to the polygons in the watershed coverage by field names which contain the grid_code of the watershed polygon. For example, the field in psurp.dbf containing data for the watershed polygon with a grid_code of 35 is 'GC35'. The same field names are maintained in the other output tables used in the surface water simulation.

With the data interpolation now complete, the surface water model can be run to simulate flow in the river basin. GH-Rivers simulates flow by generating three time series, namely Pflow, Tflow and Fflow, for each river arc and associated watershed polygon. The use of Tflow and Fflow is an important concept because it enables flow to be interpolated to any point along a given river arc. Pflow is the flow that is generated as a result of rainfall incident on a watershed polygon. It is generated from the surplus, $psurp$, by applying a fraction to account for infiltration and converting it to a runoff time series. The conversion uses a series of in-built routing functions to transform the input discrete rainfall time series to a pulse runoff time series. For a model containing dams, GH-Rivers uses a two step flow routing module to simulate the flow. The equation for the module is as follows:

$$TFlow_{i,j}^t = FFlow_i^{t-1} \bullet Llag_i + FFlow_i^t \bullet (1 - Llag_i) + PFlow_i^t - DFlow_i^t - Loss_i^t$$

where

$Llag_i$ is the normalized time lag between the from-node and to-node for a river section i and is computed as the product of the length of the section and the simulation time step, divided by the average flow velocity in the river section.

$TFlow_{i,j}^t$ is the flow passed on to downstream node, j , of the river section i

during time step t

$FFlow_i^{t-1}$ is the flow entering the upstream node of the river section i

during time step $t-1$

$PFlow_i^t$ is the flow contributed by the subwatershed of river section i

during time step t

$DFlow_i^t$ is the flow diverted from river section i during time step t

$Loss_i^t$ is the in-stream losses in river section i during time step t . It is

computed as the product of the Fflow ($FFlow_i^t$), the length of the


river section ($Length_i$) and the in-stream loss coefficient expressed

as a fraction of streamflow lost per unit length of river ($LossC_i$).

To perform a simulation of the whole basin, click on the View containing the basin coverages to make it active. Select SFlowSim from the SFlwModel menu in the View Menu Bar. An input screen containing several model parameters is presented with default values provided. A description of the model parameters is given in Table 4.3.

Parameter	Description	Default Value
Optimize	Determines type of simulation to be performed 0 → Simulation only 1 → Simulation with Parameter Optimization	0
CalPflow	Indicates whether or not to recompute pflow 0 → Do not recompute pflow 1 → Recompute pflow	0
Nresp	Number of steps of response function to be performed in converting psurf to pflow (Normal range: 6 to 12)	6
IniTmStep	Initial time step Start with 1	1
EndTmStep	Final time step Set this value to the number of records in the simulation tables (psurf.dbf, pflow.dbf, fflow.dbf and tflow.dbf). Use 12 for this model	12
ToSub	The fraction of surplus that goes to ground water (permissible range: 0 to 1)	0.1
Muskingum	Determines where Muskingum river routing procedure is to be considered. 0 → Do not perform Muskingum routing 1 → Perform Muskingum routing	0
Pltply	Determines the level of graphical display during the simulation 0 → Highlight only river arcs 1 → Highlight both polygons and arcs	1

Table 4.3: Surface Water Simulation Parameters

The results of the surface water simulation can be viewed by plotting flow profiles at various locations in the basin. To plot flows, click on the  tool in the View Tool Bar. An input screen comes up with two options of flow profiles to

be plotted. The first option allows the user to plot a longitudinal flow distribution from a user selected point to the nearest downstream outlet. The second option allows the user to plot flow time series at any location on the stream network. The user is also prompted to select one of the four flow time series (psurp, pflow, fflow and tflow) to use for the plot. Selecting one of these two flow plotting options will result in a little crosshair cursor. Click on any location in the stream network and a graph of the variation of flow rate with time at that location is generated from the selected flow time series. Clicking on points off the stream network results in a message informing the user that the selected point is not on a river arc.

4.2 CALIBRATING MODEL PARAMETERS

With the model construction phase complete, the model calibration phase can now begin. The main goal in this phase is to determine the values of flow routing parameters that result in the best match between measured and simulated runoff. An interactive optimization module is used to establish this match. It is important to note that while the parameters may vary in space, they do not vary in time. Soil, land cover or other physical conditions which may vary in time are not explicitly accounted for in the model. Instead, the effects of these conditions are implicitly accounted for by the lumped parameters. The model is calibrated over

an extended period of time to ensure that the parameters obtained are representative of average conditions in the basin.

To ascertain the quality of the runoff data provided, plots were made of the runoff and rainfall data at selected locations in the basin. Figures 4.4 and 4.5 show the respect daily rainfall and runoff recorded at Amsoul in the Souss basin.

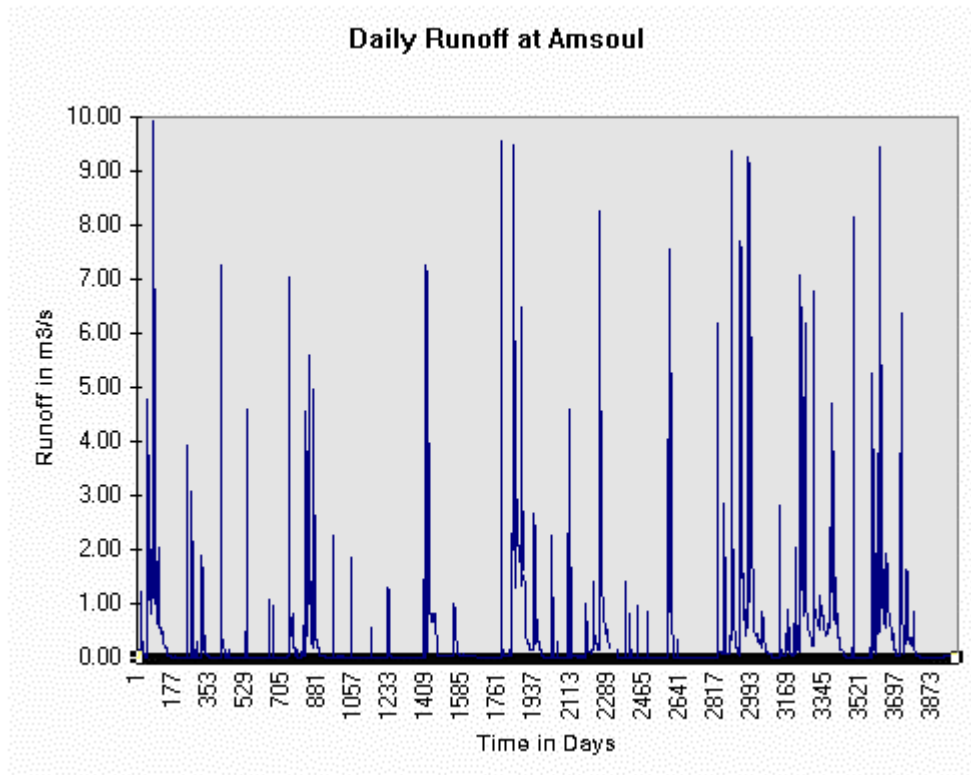


Fig. 4.4: Chart of Daily Runoff at Amsoul

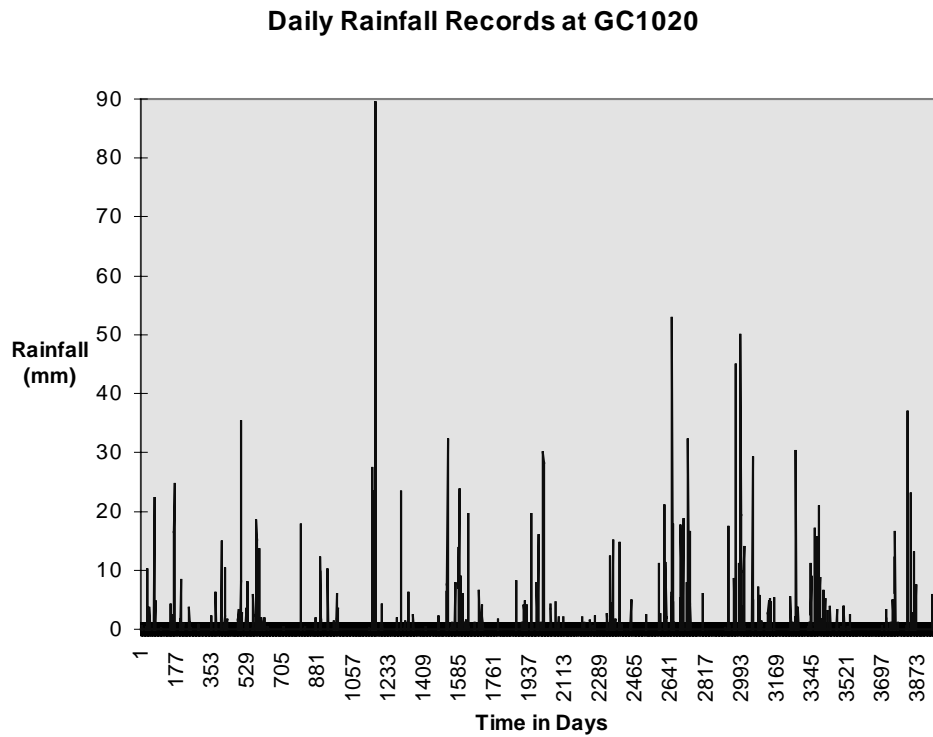


Fig. 4.5: Chart of Daily Rainfall at a Station near Amsoul

The large amount of data makes it difficult to effectively compare the trends in the two sets of data. Plotting data for shorter periods of time makes it easier to observe trends in the data. Figure 4.6 shows the monthly rainfall and runoff at Aoulouz in 1983.

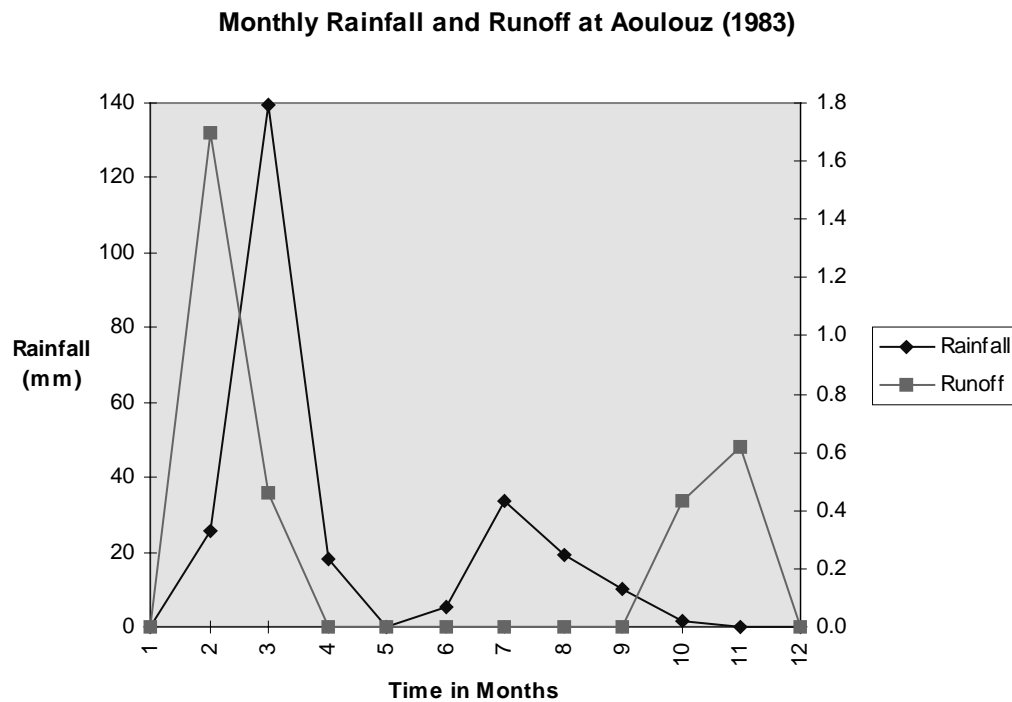


Fig. 4.6: Chart of Monthly Rainfall and Runoff at Aoulouz

The data from all the runoff stations should also show a consistent trend; high and low flows should occur consistently across the basin. Plots of runoff at five stations in the basin for 1987 are presented in Figure 4.7. Runoff data for the sixth station, Aoulouz, were not available for this year.

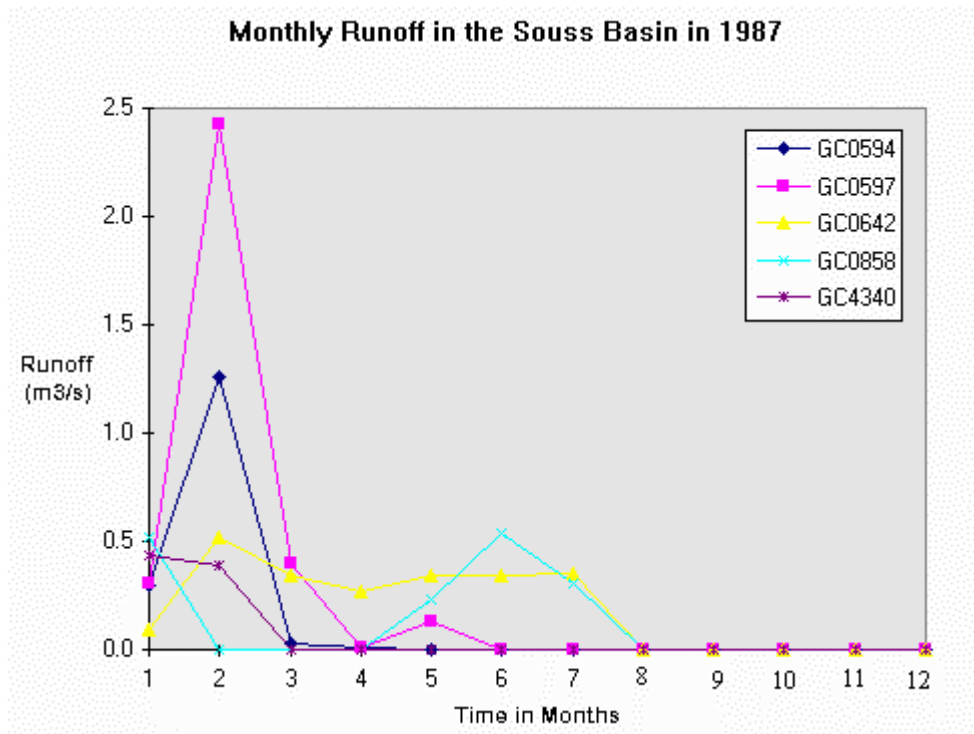


Fig. 4.7: Monthly Runoff at 5 Souss Basin Runoff Stations in 1987

Rainfall data should also show a consistent trend. Plots of monthly rainfall recorded at selected stations throughout the Souss basin are shown in Figure 4.8.

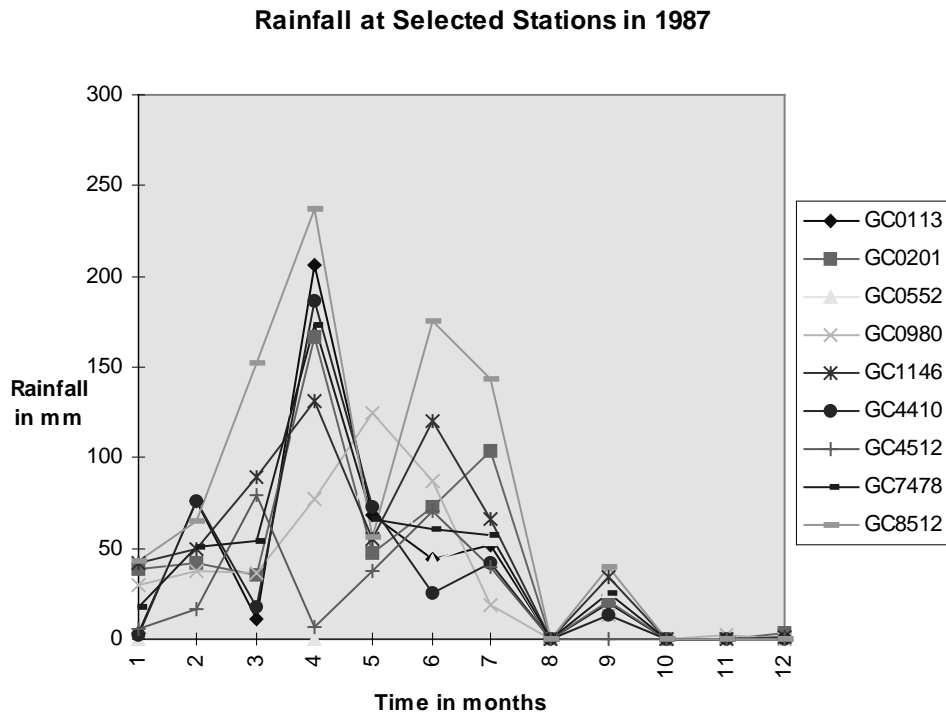





Fig. 4.8: Monthly Rainfall at Selected Souss Basin Stations in 1987

Flows were calibrated at each of the runoff stations in the basin using the interactive optimization model. The calibration period was selected by reviewing the rainfall data to select a year in which rainfall typical of the basin was recorded. After reviewing the rainfall data, 1987 was selected as the calibration year because it contained clearly defined wet and dry seasons. Flows generated by the model from the rainfall data were calibrated against runoff recorded at each of the six runoff stations in the Souss basin during that year. The distance from the runoff station to the upstream node of the arc on which the station is located, expressed as


a fraction of total length of the arc, is a required input for the calibration process. This percentage computation is performed by the flow interpolation program when interpolating flow to a flow check point. Hence by inserting a flow check point at the location of the runoff station and interpolating the flow to that point, the percentage distance along the arc can be determined. Flow check points are added by clicking on the (red flag)  tool in the View Tool Bar and choosing option 2 from the list of options provided. Clicking on top of the rain station location results in the addition of a check point. The shape file containing the check points, **flowchk.shp**, can be made visible by clicking on the little box next to the theme in the View legend. To interpolate flow to the check point, select it using the View Select  button and then click on the flow interpolation (green flag)  button. The percentage distance can be viewed by opening flowchk.dbf from the list of tables in the project window.



For each runoff station, runoff records for the calibration year are used as target flows against which simulated flows are calibrated. The optimization model obtains the target flows from the 'Target' field of the 'target.dbf' table. Hence before the optimization begins, the observed flows at the calibration station must be read into 'target.dbf'. This is done by selecting **SetTables** from under **SFlwModel** in the View Menu bar. An input screen comes up containing three

table setup options. Select option 3 (**FillOneFieldWithAnother**), which allows the user to copy the contents of a field in one table to a field of similar data structure in another table. This initiates a series of dialog boxes which allow the user to specify the target and destination fields. While the fields in the source and target tables are not required to have the same name, they must have the same data structure and a unique one-to-one relation. The table below contains the information requested and answers supplied in calibrating the runoff at Aoulouz (grid_code 0203) in the Souss basin model.

INFORMATION REQUESTED	INPUT OPTIONS	INPUT PROVIDED
Select a Target Table	(A list of tables available in the project) , OK/Cancel	Target.dbf
Select a Target Field	(The fields in the Target table selected) , OK/Cancel	Target
Select a Target Table Key Field	(The fields in the Target table selected) , OK/Cancel	Time
Select a Source Table	(A list of tables available in the project) , OK/Cancel	m87s0203.dbf
Select a Source Field	(The fields in the Source table selected) , OK/Cancel	GC0203
Select a Source Table Key field	(The fields in the Source table selected) , OK/Cancel	NTIME

Table 4.4: Requests and Inputs for Setting Optimization Target

The parameters of all the subwatersheds contributing flow to a particular runoff station are calibrated at the same time. These parameters are all attributes of the basin coverages, 'sobasin.ply' and 'soriver'. The runoff station coverage, 'hydrost', is also required to indicate the location of the runoff station being calibrated. To isolate the region draining through the runoff station, the shift key is held down and the 'sobasin.lin', 'sobasin.ply', 'soriver' and 'hydrost' themes are made active by clicking on them in the View legend. All watershed polygons and river arcs upstream of the station are selected using the View Select tool, . Such a selection ensures that only the values of parameters in the selected sections are changed when the optimization is run.

The optimization process runs much faster if separate sub-models had been created of the subwatershed polygons making up the drainage area of each runoff station. GH-Rivers contains a tool for creating a separate sub-model of selected portions of a model. Such a sub-model could have been created after selecting the portion of the model to be optimized (using the View Select  tool as described above), by clicking on the Cut  button. Selecting this button results in a series of dialog boxes prompting the user for the destination directory for the sub-model files as well as the destination View in which to display the clipped themes. Selecting the default 'cutcov' directory, the default file names

provided and the 'New View' option results in the creation of a new sub-model with a new set of clipped coverages. Such a sub-model does not need to undergo any preprocessing and can be operated on in the same manner as the parent model. However, there is at present no mechanism for merging sub-models together into a single model again in ArcView. Consequently, the values of parameters obtained in the sub-models would have had to be typed manually into the attribute tables of the coverages in the main model.

The interactive optimization module allows the user to select the parameters to be calibrated from a list of the attributes of the basin coverages. For each parameter selected, a range of values must be specified over which an optimal value will be sought. A wide range of possible values should be selected for the initial run of the optimization. The range can subsequently be narrowed down to the neighborhood of the resulting optimal values. The method of determining the optimal value must also be specified. The optimization model supports two algorithms, namely the Root Mean Square Error (RMSE) method and the Sum of Mass Discrepancies (SMD) method. The RMSE is used with parameters such as velocity which ensure a temporal match. It aims to produce a best fit by minimizing the squared difference between the target and simulated time series. The SMD is used with parameters such as the in-stream losses which ensure that the same quantity of flow is produced by the two time series. It ensures

that the sum of the differences in monthly flows is zero over the optimization period.

While six parameters may be selected simultaneously, it is unwise to select more than two or three for each run of the model. This is because an optimization with six parameters may take a long time to converge particularly when the range of possible values has not been narrowed down. The parameters used for the calibration should account for variations in both the timing and the quantity of flow. The routing parameters that can be calibrated in a given model are described in Table 4.5.

<u>Parameter</u>	<u>Description</u>	<u>Units</u>
LossC	Instream Loss Coefficient	1/km
Velocity	Instream Flow Velocity	m/s
ToRes	Fraction of Surplus Going to Subsurface Reservoir	-
PlossC	Subwatershed Loss Coefficient	1/km
ResK	Mean Residence Time In Subsurface Reservoir	T
Vfact	Overland Flow Velocity	m/s

Table 4.5: Flow Routing Parameters

For the GH-Rivers model, the overland and in-stream flow velocities influence timing while the watershed loss coefficient and the in-stream losses influence the quantity of flow. Combinations of these four parameters were tested to determine which set produced a best match. After each run, an in-built post-processing utility displays the results of the optimization in the form of a series of charts and tables.

Following are detailed instructions on how the Souss basin model was calibrated. The file containing rainfall data for 1987, `mrn87.dbf`, was added to the project in ArcView and surplus was interpolated to the centroid of the subwatershed polygons using the `CMPSURP.PRE` program as described in section 4.1 above. Surplus runoff is also a required input for the optimization program. The `soriver.shp` and `sobasin.ply` themes were made active by clicking on them in the View legend. The View Select tool was used to select the portion of the basin on which parameters were being matched as shown in Figure 4.9.

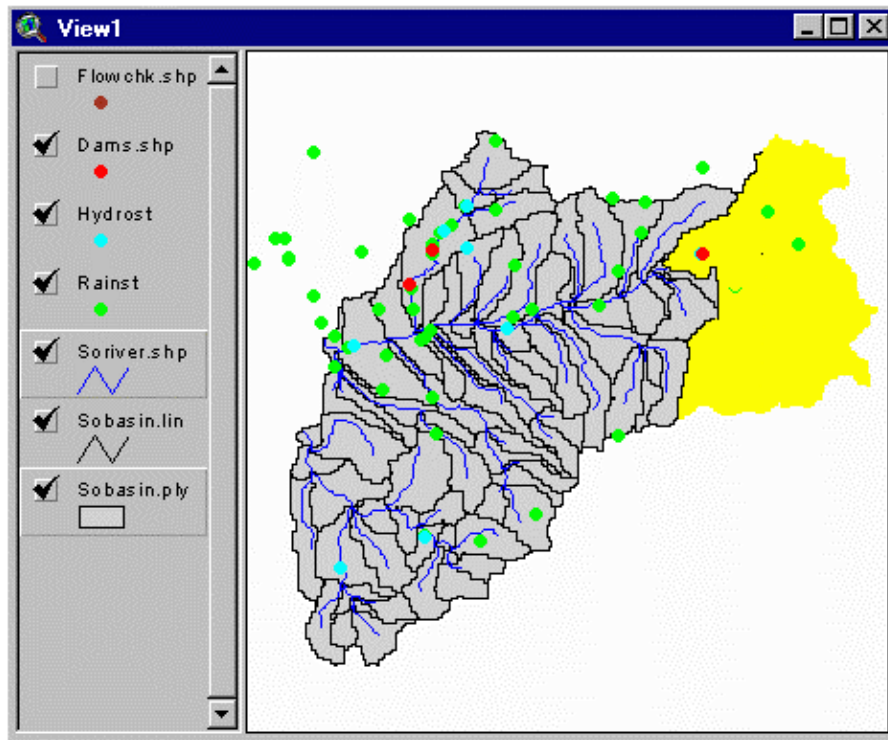



Fig. 4.9: Calibrating Flows at Aoulouz

Click on the optimization  button in the View Menu Bar to start the optimization program. Table 4.6 shows a sample of the information requested by the program and the inputs that were provided in calibrating flow at the Aoulouz in the Souss basin.

<u>Input Requested</u>	<u>Input Options</u>	<u>Input Provided</u>
Select a target theme on which results will be matched	(A list of all the themes in the active View), OK / Cancel	soriver.shp, OK
Enter ThePcnt (percentage distance along arc of the runoff station)	Permissible range: 0 → pcnt → 1 OK / Cancel	(This percentage varies depending on the runoff station being calibrated)
Select a Control file	(A list of control files), OK / Cancel Cancel to create a new control file / optctrl.ctl, OK	Select Cancel for the initial run and create a new control file, and save it as optctrl.ctl
Select themes for parameter fitting	(An input screen with a list of the active themes in the View legend), 0 / 1 0 → Do not select theme 1 → Select this theme	sobasin.ply → 1 soriver.shp → 1
Select parameters for fitting	(An input screen with a list of the parameters in the selected themes) 0 → Do not select theme 1 → Select this theme	soriver.shp, Velocity → 1 soriver.shp, LossC → 1 sobasin.ply, ToRes → 1 sobasin.ply, Vfact → 1
Confirm the list of parameters selected	(A list of the selected parameters), OK / Cancel	OK
Enter minimum value, maximum value, type of optimization (delimit with commas)	(A list of the parameters and their current values) OK / Cancel For optimization type, 0 → SMD (Sum of Mass Discrepancies) 1 → RMSE (Root Mean Square Error)	soriver.shp, Velocity: 0.1,0.5,1 soriver.shp, LossC: 0.00001,0.01,0 sobasin.ply, ToRes: 0.01,0.3,0 sobasin.ply, Vfact: 0.005,0.1,1
Confirm the selected parameters	(A list of the selected parameters), OK / Cancel	OK

Table 4.6: Requests and Inputs for Setting Up Optimization Model

The program begins to perform the optimization of the parameters selected, within the bounds set by the minimum and maximum values, using the optimization type specified. The program may not be able to find a value that satisfies all the conditions set during its first run. Several runs may be required to narrow down the range of the parameters in order to allow the model to converge. The ranges suggested in table 4.5 above were used in the Souss basin. However, there is no guarantee that the flows generated will be of the same order of magnitude as the observed flows when it converges. This is because the basin coverages contain other parameters which the model uses to perform the routing. The ideal situation would have been to calibrate a number of the other parameters listed in Table 4.7. Time limitations made it impossible to undertake such an intensive calibration effort for the Souss basin in this research. It would also have taken the focus of the study away from its primary task of reservoir planning. The parameters in the table below may be used as default values for the parameters that are not being calibrated. Using these values may enable the optimization model to converge on a solution faster. The description and units of these parameters are given in table 4.5.

Parameter	LossC	Velocity	ToRes	PLossC	ResK	Vfact
Value	0.004	0.2	0.1	0.004	6	0.013

Table 4.7: Suggested Default Values for Routing Parameters

4.3 CONSTRUCTING RESERVOIR SIMULATION

The reservoir simulation model developed is a level pool routing model. It computes the water level at the end of the month based on the storage at the beginning of the month, inflows and discharges during the month. The water level is presented in terms of the volume in storage, the surface area and the water surface elevation at the dam. Three data files are used in the reservoir simulation model. Figure 4.10 shows the relationship between these three files.

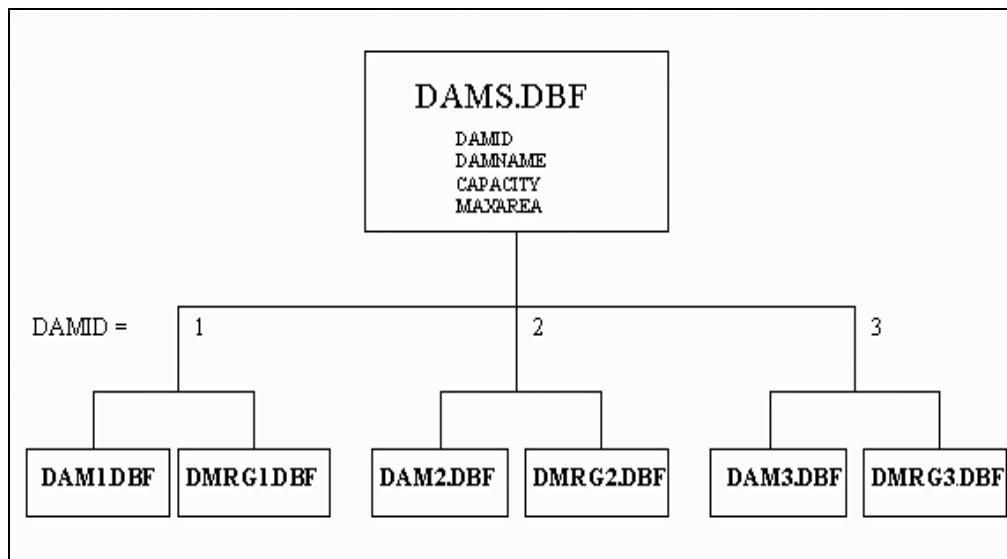


Fig. 4.10: Relation of Dam Data Files

The first of these is ‘dams.dbf’ which contains information about the dimensions and characteristics of all the dams and associated reservoirs. The

dams.dbf table for the Souss basin reservoirs is shown in Table 4.8. A description of the reservoir attributes in dams.dbf is provided in Table 4.9.

Damid	Damname	Capacity	Area	Deadst	Evt	Pdam
1	dam1	351458300	13000000.000	1458300	0.30000	0
2	dam2	244508200	7921700.000	4508200	0.30000	0
3	dam3	1525100	214600.000	25100	0.30000	2

Ndam	Storage0	Area0	Grid_code	Pcentage	Length	Upst
0	351458300	13000000	22	0.26	24313.710	351458300
3	189989900	6877000	11	0.35	58112.700	244508200
0	1141300	200000	11	0.70	58112.700	1525100

Table 4.8: Dams.dbf Table of Reservoir Attributes

The file ‘dmrg(id).dbf ‘ contains data that defines the reservoir storage-elevation and area-elevation curves. The third data file, ‘dam(id).dbf’, contains historical reservoir operation data. The model contains only one ‘dams.dbf’ file but there is a ‘dam(id).dbf’ and a ‘dmrg(id).dbf’ file associated with each of the reservoirs in the basin. The model also contains a synthetic storage-area equation which it uses, by default, to compute the surface area for reservoirs without a ‘dmrg(id).dbf’ file.

During pre-processing, an additional field is added to the attribute table of the river arc coverage, as a field called ‘HasDam’. When a dam is inserted on a particular river using the dam locator tool in the simulation program, the value of

this 'HasDam' field is set to 1. The value of the flag is increased by one every time another dam is added to the same arc. This flag triggers the reservoir simulation when the arc is encountered during the surface water simulation by calling up the reservoir simulation program, Damrt.ave. The surface water simulator also provides Damrt.ave with a list containing inflows for each time step in the period being simulated. Damrt.ave in turn establishes the characteristics of the reservoir encountered as well as initial storage and surface area from 'dams.dbf' using the damid as the key field. Reservoir operation data for the simulation time period are read from the 'dam(id).dbf' file corresponding to the reservoir encountered. Damrt.ave uses this information to compute the reservoir water balance as described below.

For a time period, i , the storage at the end of each time period, S_i , is computed from the equation:

$$S_i = S_{i-1} + I_i - R_i - W_i - L_i - E_i$$

where

S_{i-1} is the storage in cubic meters at the end of the previous time period
(initial storage for the first time period)

I_i is the inflow in cubic meters for time period i

R_i is the release in cubic meters for time period i

W_i is the withdrawal in cubic meters for time period i

L_i is the leakage in cubic meters for time period i

E_i is the evaporation loss in cubic meters for time period i .

E_i is determined from

$$E_i = NEvap_i * A_i$$

where

$NEvap_i$ is the net evaporation in meters read from the
dam(id).dbf table

A_i is the surface area of the reservoir in meters squared at
the beginning of period i

The storage computed at the end of the time period, S_i , is then compared
with the maximum reservoir storage, S_{max} , from dams.dbf.

If S_i is greater than S_{max} then

the spill is computed as the positive difference between computed storage
and the maximum storage.

$$S_{pi} = S_i - S_{max}$$

However, since the computed storage is greater than the reservoir capacity,
the storage at the end of the time period is set to the maximum storage.

$$S_i = S_{max}$$

The release is also updated to include the spill,

$$R_i = R_i + Sp_i$$

If on the other hand the computed storage, S_i , is less than the dead storage, S_{min} , specified in dams.dbf then

deficit is computed as the negative difference between the computed storage and the dead storage of the reservoir.

$$D_i = - (S_{min} - S_i)$$

Since water cannot be released from the dead storage of the reservoir, the storage for that time period is set to the dead storage,

$$S_i = S_{min}$$

The release is reduced to reflect the actual amount of water that is available for release from the reservoir

$$R_i = R_i + D_i.$$

The new storage is then used to determine a new surface area and water surface elevation by linear interpolation from the appropriate 'dmrg(id).dbf' file. The program searches through the 'dmrg(id).dbf' until it encounters a pair of successive storages with one value greater than and the other less than the new

storage. A linear interpolation is performed between these two storage values to obtain values of head and surface area that correspond to the new storage.

The initial storage and surface area in the ‘dams.dbf’ file are updated to reflect current conditions. The inflows, releases, evaporation losses, deficits and spills are also recorded in the ‘dam(id).dbf’ table for the reservoir being simulated. The fields in the dam(id).dbf tables are described in Table 4.9.

<i>FIELD</i>	<i>DESCRIPTION</i>	<i>UNITS</i>
TIME	The simulation time step, beginning with 1	none
INFLOW	Total reservoir inflow in a given month	m3
LEAKAGE	Total loss by infiltration in a given month	m3
NETEVAP	Net evaporation	m
SURFAREA	Surface area of the reservoir at the end of the month	m2
EVAPLOSS	Water lost through evaporation during the month	m3
SPILL	Water released as excess, usually as spillway overflow	m3
DEFICIT	Portion of the target withdrawal not met during the month	m3
WITHDRAW	Reservoir outflow that does not return to the river	m3
RELEASE	Reservoir outflow going into the river channel	m3
STORAGE	Total volume of water in the reservoir at the end of the month	m3
HEAD	Water surface elevation at the end of the month	m3

Table 4.9: Reservoir Simulation Fields in Dam(id).dbf Tables

The flow reaching the portion of the river downstream of the dam in each time period is also computed and returned to the surface water simulator as a list of

monthly releases. For time period i , reservoir outflow in cubic meters per second is given by:

$$O_i = R_i / (30.4 * 24 * 3600)$$

where

O_i is the outflow in cubic meters per second and

R_i is the release in cubic meters

The factor $(30.4 * 24 * 3600)$ converts the volume of water released to mean monthly flows.

If there is more than one dam on the same arc, the simulator performs the reservoir water balance for all the reservoirs on the arc before returning to the surface water simulation.

4.4 CHECKING MODEL RELIABILITY

After the parameters influencing flow have been calibrated, the model is tested to determine the reliability of generated flows. An extended period including several seasonal cycles should be chosen for this simulation. For the Souss basin, the model was simulated with 60 records. For this run, the number of records in each of the data files used for routing flows in GH-Rivers was increased to 60. GH-Rivers contains a menu item called **AddRecords** which can be used for

adding empty records to data files, one file at a time. It is however quicker to set up a new model using the modified preprocessing procedure described in Figure 4.11.

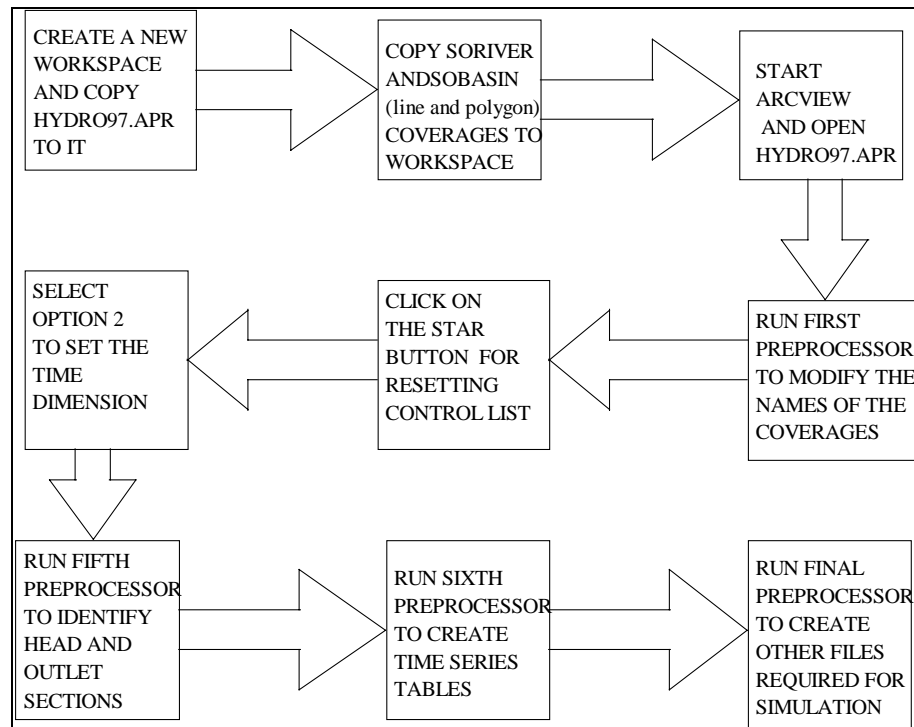


Fig. 4.11: Flow Diagram of Modified Preprocessing Procedure

To perform the modified preprocessing run, the processed coverages and their attribute tables as well as the reservoir regulation files, 'dam(id).dbf', are copied to a new workspace. The button is selected from the view menu bar. This opens up an input screen with three options. Selecting option 2, **Set Time**

Dimension, results in a new input screen which enables the user to enter the initial time step (**IniTmStep**) and the final time step (**EndTmStep**). These parameters were set to 1 and 60 respectively for the Souss Model. The fifth, sixth and seventh pre-processing steps were then run as before to create a new model with 60 records. This modified approach ensures that model parameters that have already been calibrated are retained as attributes of the basin coverages.

Reservoirs are generally constructed on rivers when the storage requirements exceed existing storage capacity. Hence different reservoirs tend to be constructed at different times during the development of the river basin. Instances may occur when a reservoir which was not yet constructed at the beginning of the simulation period becomes operational later on during the simulation. The problem of how to handle reservoirs which are only online for portions of the simulation period was encountered when setting up the model for the extended period in the Souss basin. A 60 month model set up for the period 1990 - 94 indicated that at the beginning of the period, only the reservoir at Dhkila was operational. The other two reservoirs became operational later during this period. This problem was solved by setting the storage in all the reservoirs to capacity at the beginning of the simulation period. With reservoir withdrawals as well as evaporation and leakage losses set to zero, any inflow caused the storage capacity to be exceeded for periods when the reservoir had not yet been

constructed. Hence all inflows are released as spills allowing flow to proceed downstream as if there was no reservoir. The effects of the actual reservoir operations are introduced by altering the withdrawals and losses for periods when the reservoir subsequently becomes operational.

Observed and simulated flows for this period can be compared at one of the runoff stations in the basin. Measured flows at the selected runoff station should be extracted into a separate file. This file is used to fill the target.dbf table against which simulated flows are compared. It is important to note that while values of simulated flow can be obtained for any location in the basin, the target flow is only correct at the runoff station where the measurement was made. The surface water simulation can now be run for the extended period. After the run, a plot of the observed and simulated flow against time can be made for the target station. The percentage deviation of the simulated values from the observed values are presented in the flowtime.dbf file. The RMSE and the SMD for the simulation period are also presented in the flowdist.dbf file. Statistical methods may also be used to define the reliability of the model in terms of measures such as the standard deviation, the standard error or the correlation of individual flows. These are however not built into the model. The data would have to be extracted into a spreadsheet or other statistical analysis software for computation. The results of the parameter calibration process are described in section 5.1 of this report.

4.5 USING THE MODEL AS A PLANNING TOOL

The traditional approach to reservoir simulation involves representing the reservoirs as a set of independent units. Rivers are represented as lines linking the reservoirs. The primary limitation of this approach is that changes in the system require the reconstruction of the entire model. The main advantage of including reservoirs in a map based simulation model is that various system and operation options can be examined without reconstructing the model. The number and geographical location of reservoirs can be altered to examine the effects of location decisions. The flows resulting from various operation and demand alternatives can also be examined at any point in the river network. To improve the utility of the model as a planning tool, some typical reservoir planning problems were examined. This led to the definition of additional features of the simulation program that a planner would require in using the model to study possible solutions scenarios.

The first problem involves checking the ability of an existing reservoir to meet the needs of a new project. For this problem, the projected mean monthly demands of the new project must be estimated and added to the historical demand data in the dam(id).dbf. The planner may want to simulate a period, say 10 years, to examine the performance of the reservoir during that period. If the model was being used to simulate flow over such an extended period of time, a tool would be

required to enable the user to insert a set of 12 monthly values only once instead of entering the same set of values 10 times. A tool was developed for adding such repetitive data into the dam(id).dbf tables containing reservoir operation data. With this tool, flows for a number of years can be generated using the projected mean monthly demands.

Over such a simulation period, the planner would want to determine the number of times the reservoir level falls outside its normal range. The reservoir level is said to have gone outside its normal range when it either falls below the dead storage level or exceeds its maximum capacity. A deficit and a spill would be recorded in the respective cases. This information is supplied in the simulation program by a utility that generates a summarized table of reservoir performance.


Mean flow rates would also need to be monitored at points downstream of the reservoir. The flow check object performs this task. Flow check points can be inserted along any arc in the stream network to allow flow to be interpolated to that point. The flow plotting tool also enables flow time series to be plotted at any point in the stream network.

The second problem involves examining the effects of installing a new storage reservoir. In basins with high seasonal variations, large volumes of water

must be stored during high flow periods for use in low flow periods. Existing reservoirs may be insufficient to store the volume of water that becomes available during the high flow periods. Excess flow must be released to the channel when there is insufficient room to store incoming flows. Possible solutions to this problem include constructing a new reservoir upstream or downstream of an existing one to store some of the excess flow. The dam at Abdel Moumen plays this role for the dam at Dhkila in the Souss basin.

The first task in seeking a solution to this problem is to quantify the volume and distribution of excess flow. A calibrated model could be used to perform the initial analysis to quantify the excess flow by running the simulation with historical rainfall time series. Setting up the model as in the first case would result in spills being recorded whenever maximum storage levels are exceeded. The spills could be summed up as a measure of the total excess flow over the simulation period. With the total excess flow known, an annual excess flow could be calculated and used as the storage capacity to be provided for a new reservoir. A dam could be inserted either upstream or downstream of the existing reservoir and the simulation run again to determine if it is sufficient to hold the excess flow. For this task, the simulation model would require tools for inserting and removing single dams as required.

New reservoirs may also be installed on rivers to regulate flow reaching downstream sections. One such reservoir is located at Aoulouz in the Souss basin to promote infiltration in the aquifer recharge zone by regulating flow. The problem in this instance is that releasing too much water may cause excess water to flow out of the basin. On the other hand, reservoir storage capacity and other considerations such as excess losses by evaporation may limit the amount of water that can be stored in the reservoir. Planners are faced with the task of determining how much water to release to ensure minimum outflow from the basin.

The simulation model allows the user to install reservoirs and flow check points at various locations. A tool which allows the monthly release schedule to be updated was developed. By allowing the user to define the how much water is released during each simulation time period, the model allows the user to study the effect of various regulation schedules. These effects can be examined by monitoring the flow in various river reaches in the basin using the flow time series plotting tool, . Long term operation policies can be determined after studying the effects of various release schedules.

In addition to constructing new reservoirs, existing ones may also have to be removed. A number of the dams constructed in the peak construction period in the 1960's are due for decommissioning. Poor planning practices have also


resulted in unnecessary dams which may be adversely affecting flow in many river basins, particularly in developing countries. It is expected that the removal and deregulation of reservoirs will continue to become increasingly common. Planners must be able to provide answers as to the hydrologic consequences of removing an existing dam.


The model allows the user to remove an existing dam and examine the effects on storage in other reservoirs and downstream river flow in the basin. The method of removing reservoirs in the original model involved using the dam locator tool to insert a new set of reservoirs. In this approach, the dams are removed from the 'dams.dbf' table and the 'Hasdam' attribute is set to 0 but the associated 'dam(id).dbf' and 'dmrg(id).dbf' tables are retained. The new dam removal tool developed allows the user to remove a single dam and its associated dam(id).dbf table. The simulation can be run with the remaining set of dams to examine the effects of not having the dam in place.

4.6 DEVELOPING THE DAMS EXTENSION

Examining the scenarios above provided insight into possible uses of the simulation model and additional tools that were required to enable it to be used more effectively in reservoir planning. An ArcView extension was developed in

the object oriented programming language, Avenue, to enable the user to analyze the reservoir planning scenarios described above. The extension, called DAMS, operates on the same theoretical basis as the main Souss basin model. However, it contains menu driven tools which enable the user to interactively add or remove reservoirs. Its pre-processing programs also allow the operation data and initial storage levels in the reservoirs to be altered. A post-processing program allows the user to generate a summary of simulation results. The functions of the various menu items in the DAMS extension and operating instructions are provided below. The setup of the dam at Aoulouz in the Souss basin is used as an example for illustrating the functionality of DAMS.

The dams in the **bigdams.shp** coverage are not recognized by the model during simulation; they are merely used as a guide to mark the location of the existing reservoirs in the basin. Dams inserted using the program tools are stored in the **dams.shp** coverage. The original Dam locator tool that was created for GH-Rivers is activated by selecting the Pennant Red icon in the View Tool Bar. The user is then prompted to specify the type of object to be inserted. Options include flow check points, diversion objects and dams. If option 2 (insert dams) is selected, a second message box appears, prompting the user to specify whether to start a new set of objects or to append to the existing set. As shown in Figure 4.12, the DAMS menu contains an item called **Insert Dams** which activates, , the

Dam locator tool in the View Menu Bar. However, instead of presenting the user with the options usually presented by the  tool, Insert Dams presents a little crosshair cursor which allows the user to immediately start inserting dams.

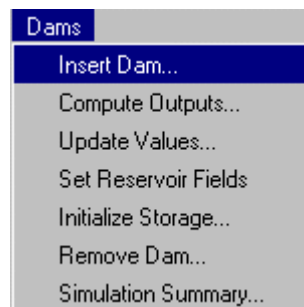




Fig. 4.12: Inserting Dams from the DAMS menu

After choosing the **Insert Dams** option, click on the location in the View, where the dam is to be inserted and an input table with a list of parameters appears. Table 4.10 contains a list of the parameters requested, their significance and the values assigned to the parameters in setting up the dam at Aoulouz.

<u>Parameter</u>	<u>Description</u>	<u>Default</u>
Damid	Identification number assigned to the dam.	1
Damname	The name of the dam. Use the main name of the dam(id).dbf file as the 'damname'.	Dam1
Capacity	Maximum reservoir storage in cubic meters. Releases are made in the form of spills when this storage is exceeded.	351458300
Area	Maximum surface area of the reservoir in meters squared.	13000000
Upst	Upper or live storage in cubic meters. Refers to the amount of water that is available for release.	350000000
Deadst	Dead storage in cubic meters. Represents the minimum storage level required to make releases.	1458300
Evt	Monthly evaporation in meters. The default value of this parameter can be used since evaporation data are provided in the dam(id).dbf tables	0.3
Pdam	Damid of the previous dam upstream dam on the same river arc.	0
Ndam	Damid of the next dam downstream on the same river arc.	0
Storage0	Initial reservoir storage in cubic meters	351458300
Area0	Initial reservoir surface area in meters squared.	13000000

Table 4.10: Setting Up Dam(id).dbf Tables

As with the  dam locator tool, a dam(id).dbf file is created for each dam inserted into the model. This file serves as the output file for reservoir simulation results.

The DAMS extension also contains a menu item for removing selected dams from the model. Highlight the dams to be removed using the View Select  tool. Select **Remove Dam** from the **Dams** menu of the View Menu Bar as shown in Figure 4.13.

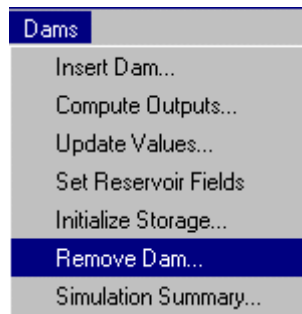


Fig.4.13: Removing Dams from the DAMS menu

A prompts comes up for the user to confirm the removal of the selected dams and associated dam(id).dbf files. Click YES to confirm the removal. Insert a dam at Aoulouz by clicking on top of the location marked by the dam in the bigdams.shp coverage.

As described in section 4.1, GH-Rivers contains a synthetic equation that is used as a default to determine the relationship between reservoir storage and surface area. However, reservoir regulation tables should be supplied if such data is available. The reservoir regulation tables contain data for establishing the relationship between reservoir storage, surface area and head. For the tables to be used in the model, their file names and field names must conform to the standard specified in developing the model. As shown in Figure 4.14, DAMS contains a menu item called **Set Reservoir Fields** which creates a new table compliant of format from input regulation tables.

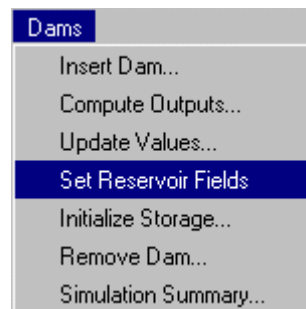


Fig. 4.14: Setting Reservoir Fields from the DAMS menu

Selecting **Set Reservoir Fields** from the DAMS menu initiates a dialog to identify the ‘damid’ of dam for which the regulation data is being processed and the input file containing the regulation data. It also asks the user to specify **a key field, a volume field, an area field and an elevation field**. The actual field names in the inputs tables may vary but the fields must be numerical with values in SI

units. The **Set Reservoir Fields** utility creates a new table with file name **dmrg(id).dbf** and adds it to the ArcView project.

The reservoir simulation model is able to account for monthly variations in demand, losses and releases. Inflows are generated by the surface water simulation and do not have to be provided as inputs. The variable inputs can be entered by editing the dam tables directly. However, this may not be desirable particularly when running a simulation for multiple years. Because of variations in the purposes served by reservoirs, the fields in the operation tables are different. The data in these tables must now be read into the standard reservoir operation tables, **dam(id).dbf**, created by the model. The key point to understand in setting up the operation data is that while there is only one class of inputs (**Inflow**) to the reservoirs, there are three classes of outflow. The three classes of **outflow** include **withdrawals**, **losses** and **releases**. **Withdrawals** are those flows that are diverted from the reservoir for other uses and do not return to the river channel. Flows to meet irrigation and water supply needs fall in this category. **Releases** on the other hand are flows which are taken out of the reservoir but which are put back into the river channel and are subsequently available for use further downstream. Spills, minimum flow releases and, in the case of Aoulouz, releases to the downstream aquifer fall in this category.

Losses are those flows that are neither diverted for other uses nor released back into the channel. Evaporation and leakage fall into this category. All withdrawals are represented in the model by the **Withdraw** field of the **dam(id).dbf** table. Releases can be entered directly into the **Release** field of the **dam(id).dbf** file. Where there are multiple withdrawals or releases, the **Compute Outputs** utility program from the DAMS menu may be used for summing up the components and updating the values in **dam(id).dbf**. Losses are represented by the **Leakage** and **Netevap** (net evaporation in m) fields in the same table. The dam fields from which inputs are read during simulation include two natural processes, leakage and evaporation ("**Leakage** and **NetEvap**"), and two controlled processes, withdrawal and release ("**Withdraw** and **Release**"). Hence subsequent runs of the model to examine reservoir management options should only involve alterations to the two controlled processes. A utility is provided in the DAMS menu for making such alterations.

For each reservoir in the model, classify the outputs in terms of releases, withdrawal and losses. If there are multiple releases or withdrawals, then use **the Compute Outputs** utility, shown in Figure 4.15, to sum up the various components into a single release or withdrawal per month. Losses do not pose a

problem since the sources of losses, leakage and evaporation, are common to all reservoirs and are handled separately in the dam(id).dbf tables.

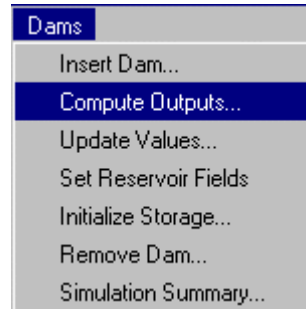


Fig.4.15: Computing Outputs from the DAMS menu

Selecting **Compute Outputs** initiates a dialog to specify the dam to be edited, the type of outflow to be computed (withdrawal or release) and the source of the outflow data. These outflows are stored in the **release(id).dbf** and **demand(id).dbf** files created at run time for releases and withdrawals, respectively.

The monthly aquifer releases are now updated in the temporary table for storing release data for dam 1, **release1.dbf**. The user is prompted on whether or not to continue updating more fields. Select **Yes** and enter values from the **release** field of **aouldmop.dbf** into the **release2** field of the **target** table. Again, the values will be updated in **release1.dbf**. Now, select **No** to stop entering more values. The

net monthly release will be computed and stored in the **total** field of **release1.dbf**. Another prompt allows the user to determine whether or not to enter the net release values into the **dam1.dbf**. Select **Yes** to update the values in **dam1.dbf**.

The withdrawal, leakage and evaporation can be entered directly into the **dam1.dbf** tables. To begin entering the month values directly into the tables, select the **Update Values** utility under the **Dams** menu of the View Menu Bar as shown in Figure 4.16.

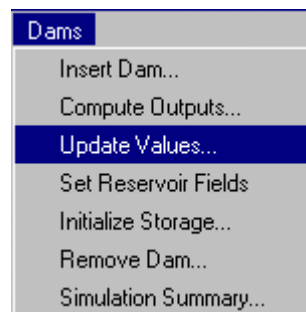


Fig.4.16: Updating Values from the DAMS menu

The user is asked to select the dam to be updated (dam 1) and the source table (**aouldmop.dbf**). Another prompt comes up which allows the user to select a **source field** followed by a **target field** in **dam1.dbf**. Input the source and target fields shown in Table 4.11.

<u>Source Field</u>	<u>Target Field</u>
Irrigation	Withdraw
Leakage	Leakage
Evaporation	NetEvap

Table 4.11: Source and Target Fields for Setting Up Dam(id).dbf

The program allows the user to update any of the fields in **dam1.dbf** (including the already updated release field). After each update, the user is prompted on whether or not to update more fields. Continue to select **Yes** until all three input fields in dam1.dbf have been set up.

It may sometimes be necessary to run the simulation model under a different set of initial conditions other than those specified during the creation of the model. To vary the initial reservoir storage, select '**Initialize Storage**' from under the **Dams** menu of the View Menu Bar and select the edit dam when prompted to do so. The input screen shown in Figure 4.17 appears for the user to enter the initial storage. Entering a storage and clicking OK causes the initial storage to be updated in dams.dbf.



Fig.4.17: Input Screen for Initializing Storage

After setting up the reservoir simulation tables, perform the surface water simulation as usual. At the completion of the simulation run, the user has the option of opening up the dam(id).dbf files to view the results for each individual reservoir on a monthly scale. However, a better assessment of the performance of the reservoirs can be obtained by generating a summary of the results. Select **Simulation Summary** from the DAMS menu, as shown in Figure 4.18, to generate a table containing summarized statistics of the performance of each reservoir. The fields in the table, called summary.dbf, are described in Table 4.12.

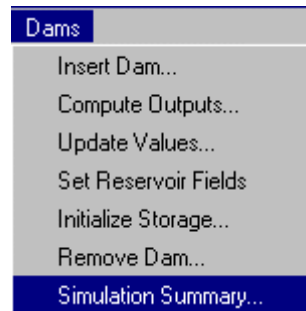


Fig.4.18: Generating a Simulation Summary from DAMS menu

<u>Field</u>	<u>Description</u>
Damid	the dam identification number
Totalmon	the number of months simulated
Deficitmon	the number of months in which deficits were recorded
Spillmon	the number of months in which spills were recorded
TotalInflow	net inflow generated during the simulation period
TotalDefici	net deficit for the simulation period
TotalSpill	net surplus inflow released as spill
Evap%	the percentage of total inflow lost through evaporation
Leak%	the percentage of total inflow seeping out of the reservoir
Release%	the percentage of total inflow released back into the river
Withdraw	the percentage of total inflow that is diverted for other uses
ChngStor%	the change in reservoir storage as a percentage of the initial

Table 4.12: Reservoir Simulation Summary Fields in Summary.dbf

Chapter 5: Simulation Results

5.1 PARAMETER CALIBRATION RESULTS

The surface water simulation was calibrated at daily and monthly time scales. The calibration was performed using four parameters namely, the overland flow velocity ('Vfact'), the watershed loss coefficient ('ToRes'), the in-stream flow velocity ('Velocity') and the river loss coefficient ('LossC'). The optimization procedure was run several times using various combinations of these parameters. For both the daily and monthly simulations, the combination of in-stream velocity and river loss coefficient were shown to produce best results. These two parameters were optimized with the overland flow velocity and the watershed loss coefficient set to 0.013m/s and 0.1 respectively. The values of the parameters obtained at the two time scales are presented in Table 5.1.

Parameter	Daily	Monthly
Calibration Period	365 days	12 months
RMSE	0.987	0.828
SMD	-0.00412	-0.0471
In-stream Velocity (m/s)	0.02	0.209
River Loss Coefficient	0.9	0.00685
time to converge	3 to 12 hours	0.25 to 1 hour

Table 5.1: Comparison of Daily and Monthly Parameters

The values presented are a measure of the total deviation of the simulated flows from the target values during the optimization period. The graphs presented in Figures 5.1 and 5.2 illustrate the nature of the match between individual flows at daily and monthly time scales. Both sets of graphs represent readings for the first 12 simulation time steps taken at Ibergnaten, just upstream of Aoulouz.

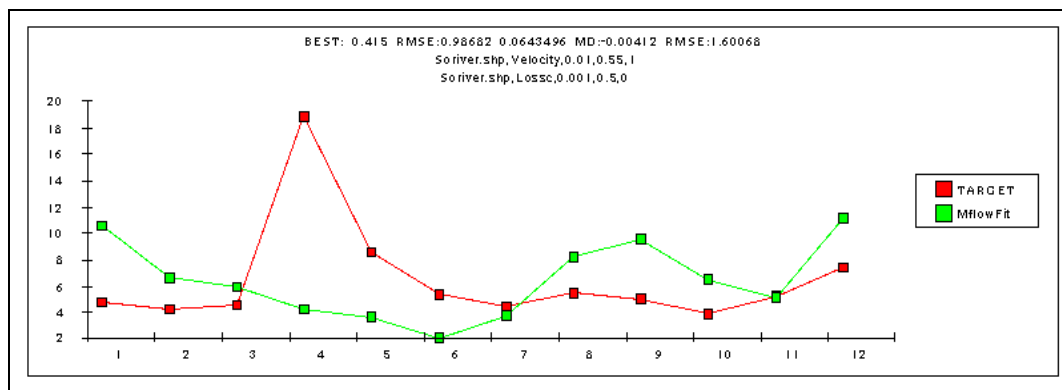


Fig. 5.1: Measured (Target) vs. Simulated (Mflowfit) Daily Flows

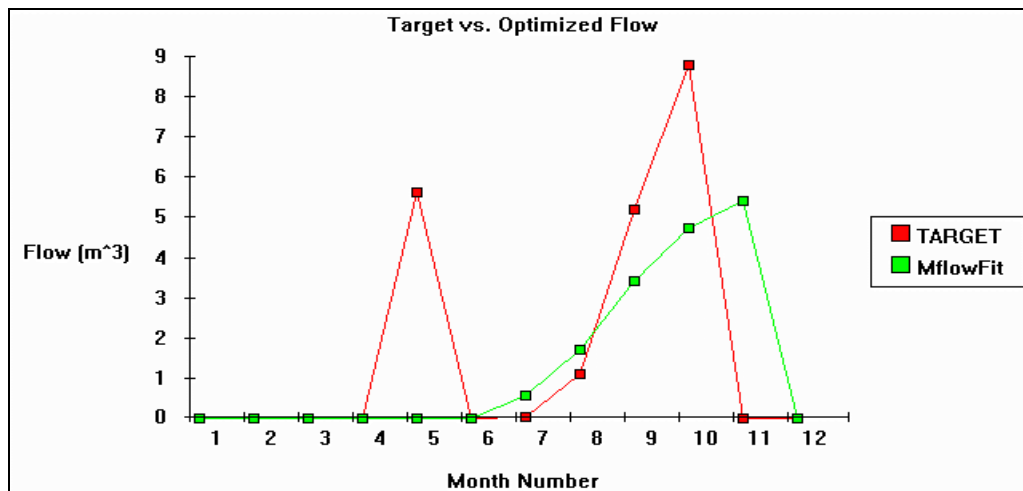


Fig. 5.2: Measured (Target) vs. Simulated (Mflowfit) Monthly Flows

The effect of spatial scale can also be observed by comparing the results of simulations at two different spatial scales. A comparison of the results obtained in the Souss basin with those obtained in the larger Niger basin provides some insight into the performance of the model at various spatial scales. The results presented in the table below provide a comparison of the parameters obtained in the two basins.

Parameter	Souss Basin	Niger Basin
Station Name	Ibergnaten	Koulikoro
Basin Drainage Area (km ²)	19000	2337000
Station Drainage Area (km ²)	2445	120000
Calibration Period (months)	12	90
RMSE	0.828	0.32
SMD	-0.0471	0.0000116
In-stream Velocity (m/s)	0.209	0.192
River Loss Coefficient	0.00685	0.000985

Table 5.2: Comparison of Souss and Niger Basins Parameters

5.2 OBSERVED VERSUS SIMULATED FLOWS

The match obtained during the simulation period is not a sufficient measure of the utility of the model. A calibrated model should be able to generate reasonable flows for a wide range of rainfall scenarios outside of the calibration period. There is no basis for comparing flows generated along river reaches where

there was no runoff station. Hence, the accuracy of flows generated within such reaches cannot be ascertained. Flows generated at runoff stations can however be compared with observed runoff. The graphs presented in Figures 5.3 through 5.5 show the nature of the match between observed and simulated flows at three stations where flow had previously been calibrated. Simulated flows are labeled as SMFLOW while observed or measured flows are labeled as MSFLOW. The data includes monthly flows for 1982 (Figure 5.3) and 1992 (Figures 5.4 and 5.5), both periods outside of the calibration year. The bars in the graphs represents mean monthly flows, beginning from January.

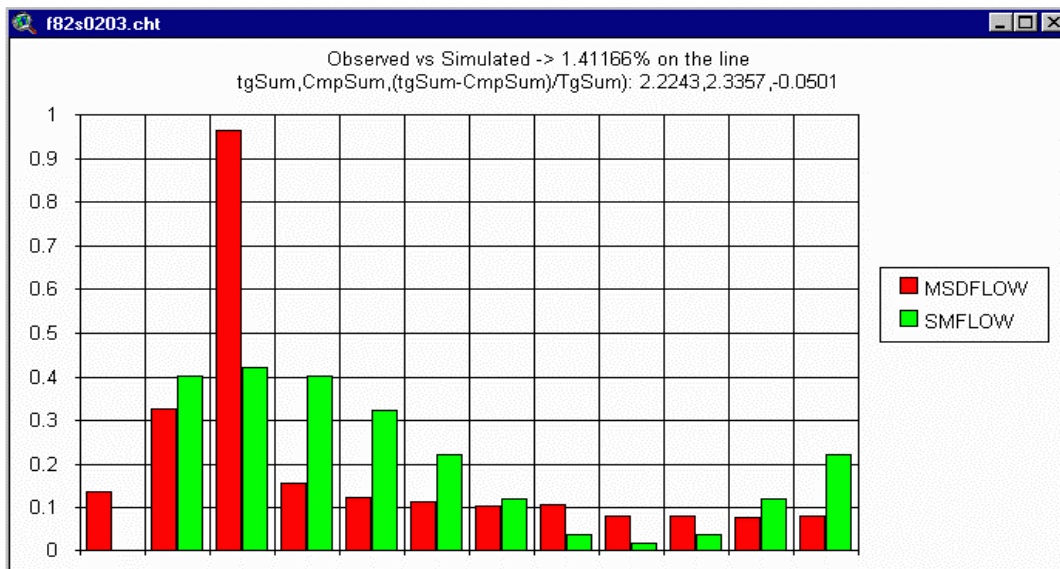


Fig. 5.3: Measured vs. Simulated Flows at Aoulouz, 1982

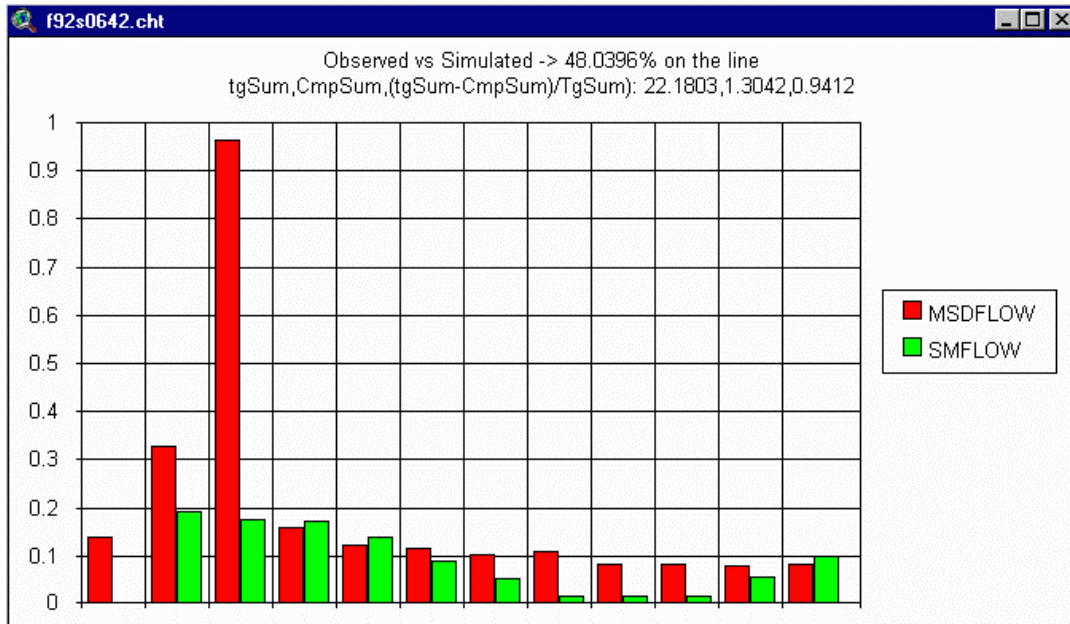


Fig.5.4: Measured vs. Simulated Flows at Immerguen,1992

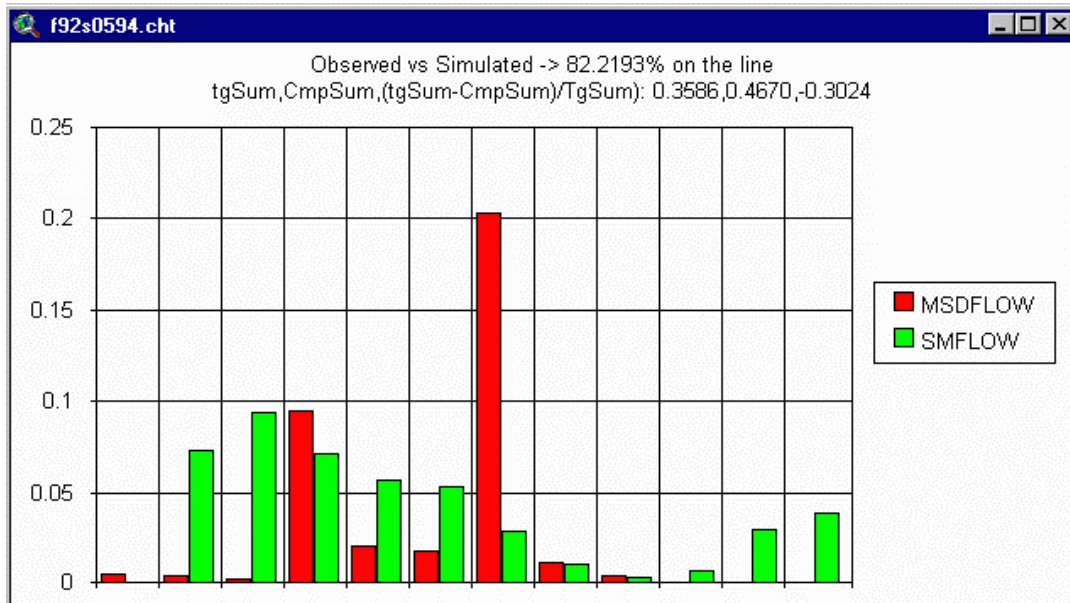


Fig. 5.5: Measured vs. Simulated Flows at Amsoul, 1992

The flows generated by the model tended to vary gradually. To check its performance under irregular flow conditions, observed and simulated flows were compared during a period of highly variable flows. The graphs presented in Figures 5.6 and 5.7, illustrate the nature of the match obtained at two such stations. It is worth noting that the measured flows at these stations are not consistent with the kind of trend that would be expected for unregulated flow data.

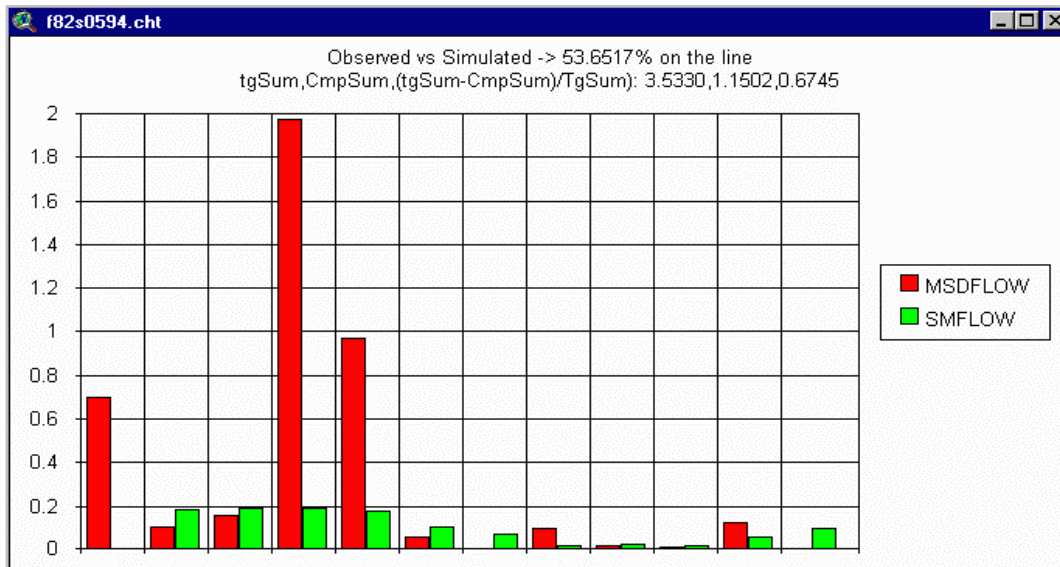


Fig. 5.6: Measured vs. Simulated Flows at Amsoul, 1982

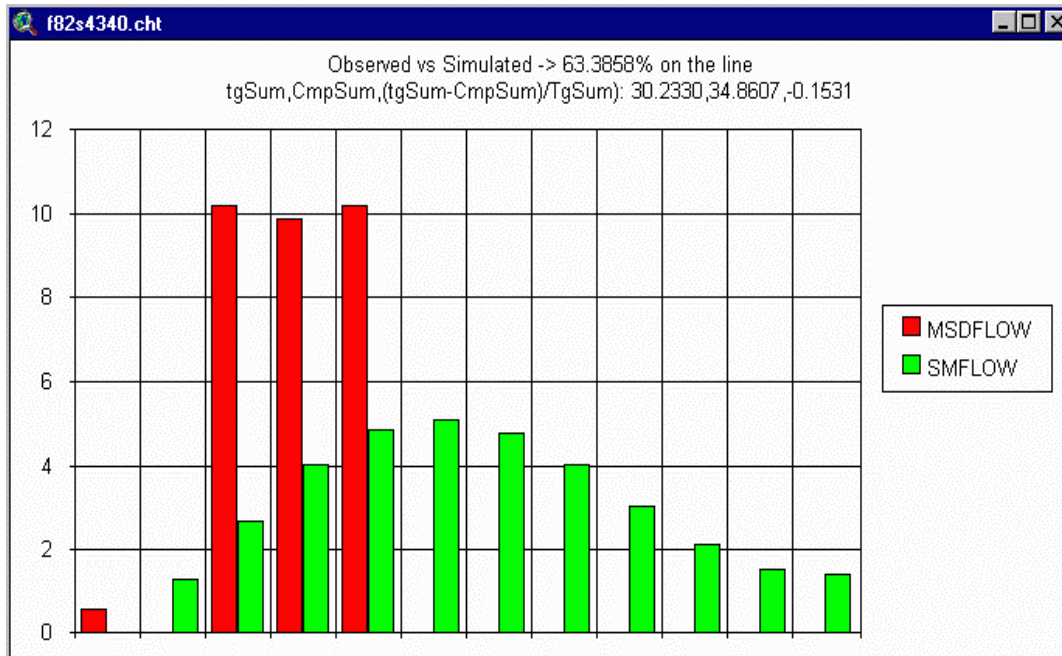


Fig. 5.7: Measured vs. Simulated Flows at Ait Melloul, 1982

5.3 RESERVOIR SIMULATION RESULTS

As described in section 5.2 above, the reservoir simulation model can be used to answer a wide range of planning scenarios. One such scenario was analyzed to demonstrate the utility of the model. The model was reconstructed with 60 monthly records to cover a five year period running from August 1989 to July 1994. August was chosen as the first month of the simulation period because records of reservoir storage levels are available for August 1st each year. A simulation was performed using the inflows and outflows specified in the reservoir operation tables for each of the three reservoirs as inputs. A summary of the

simulation was generated using the **Simulation Summary** option in the DAMS menu and pie charts were plotted from the resulting summary.dbf tables. These charts are presented in Figure 5.8.

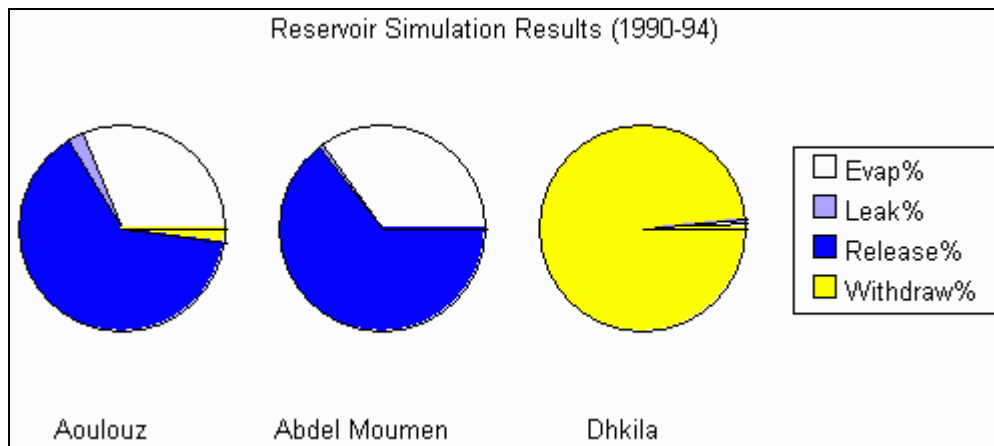


Fig. 5.8: Reservoir Simulation Results, 1990-94

The pie charts highlight the differences in the operation of the three reservoirs. Aoulouz and Abdel Moumen both record evaporation losses amounting to about 30% of inflow. Dhkila on the other hand records almost no evaporation losses since most of the water entering the reservoir is immediately diverted for other uses. Consequently, almost all inflow is converted to withdrawal resulting in little net change in its storage. Aoulouz also records some withdrawal but Abdel Moumen which acts as the storage reservoir for Dhkila, records no withdrawals.

The performance of the simulation model in predicting reservoir storage was assessed by plotting the computed storage against the storage stated in the reservoir operation tables. Figure 5.9 shows the variation in storage at Abdel Moumen during the simulation period.

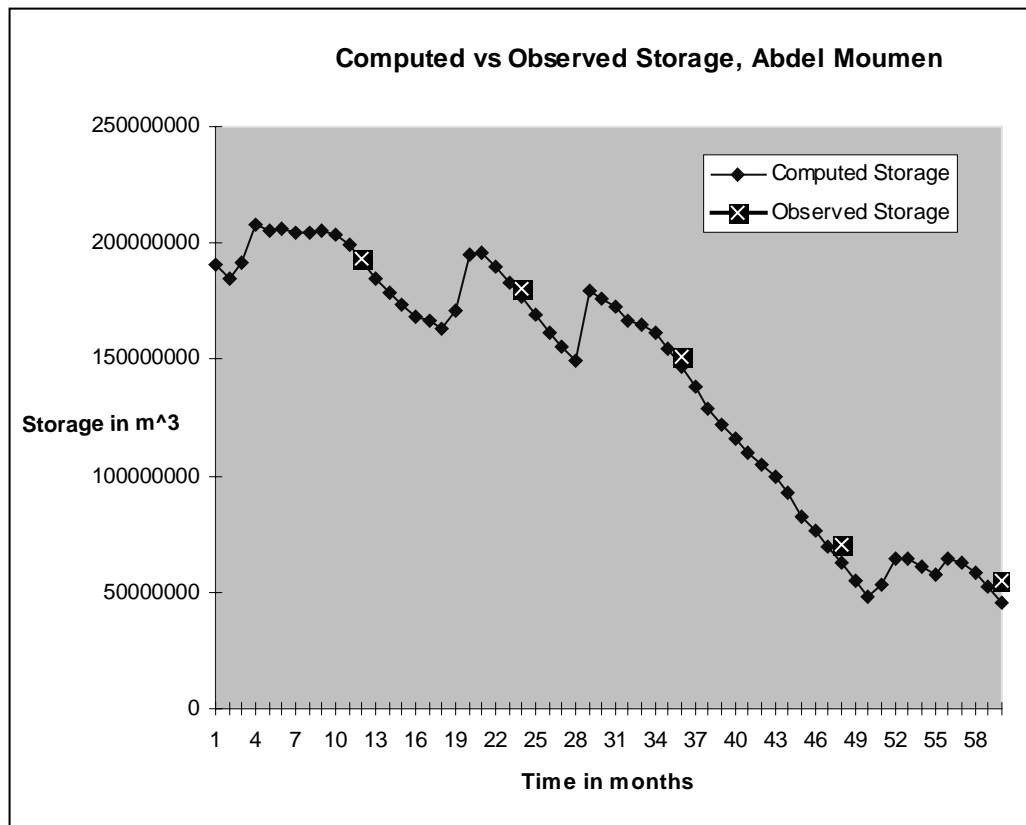


Fig. 5.9: Variation of Reservoir Storage at Abdel Moumen

The storage values generated by the model were very close to the observed storage. The computed storage values are slightly less than observed storage. This

difference can be attributed to the difference between the computed and observed evaporation losses. As shown in Figure 5.10, the evaporation losses computed by the model are slightly higher than those stated in the reservoir operation tables.

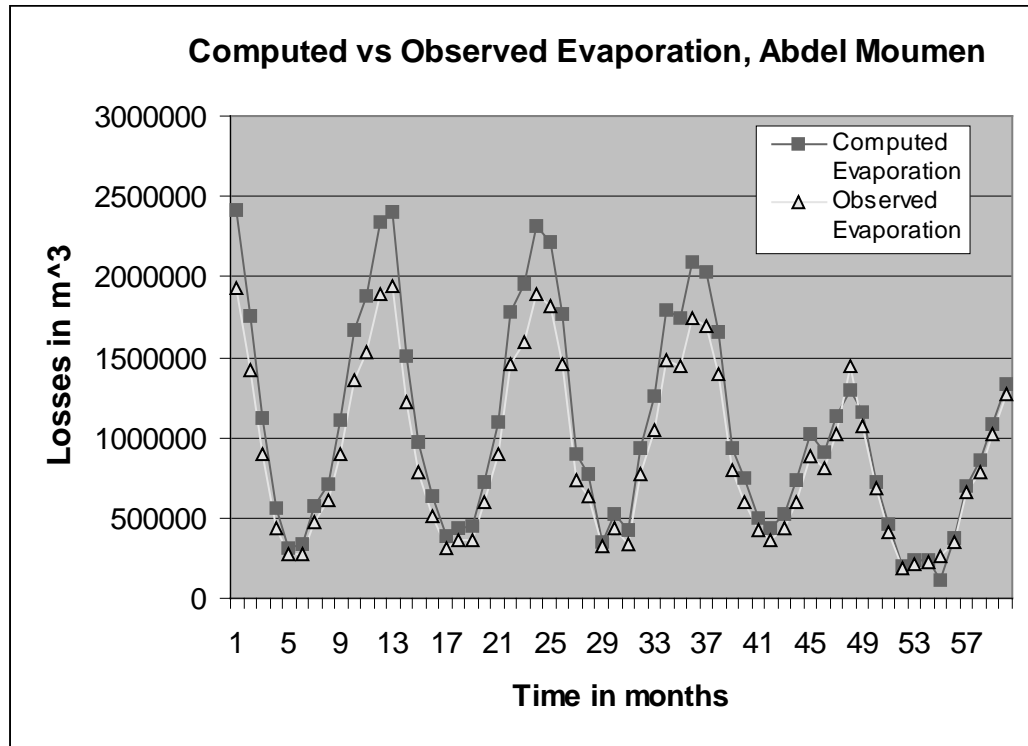


Fig. 5.10: Observed vs. Simulated Evaporation Losses at Abdel Moumen

The evaporation losses are calculated based on the surface area of the reservoir corresponding to the mean of storage at the beginning and end of each month. The storage at the end of the month is arrived at by using the storage at the beginning of the month to estimate reservoir surface area and hence evaporation losses. This storage is then used to calculate the mean surface area for the month.

The iteration of estimating surface area and storage is repeated 10 times to allow the process to converge. Figure 5.11 shows the variation of reservoir surface area with time for the reservoir at Abdel Moumen.

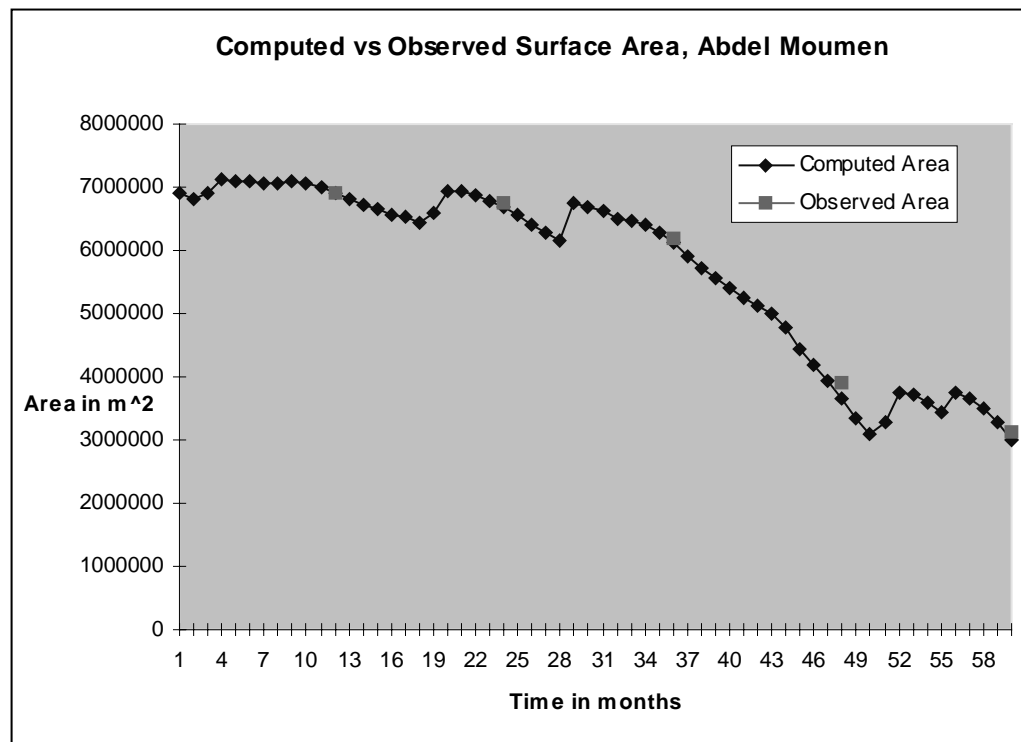


Figure 5.11: Variation of Reservoir Surface Area at Abdel Moumen

While the iteration process significantly reduced the difference between the computed and observed evaporation, the cumulative effect of this difference results in the closure error observed in the storage curve.

A comparison of the observed and simulated water surface elevation at Abdel Moumen is shown in Figure 5.12. The graph reveals the same trend as the graph of storage.

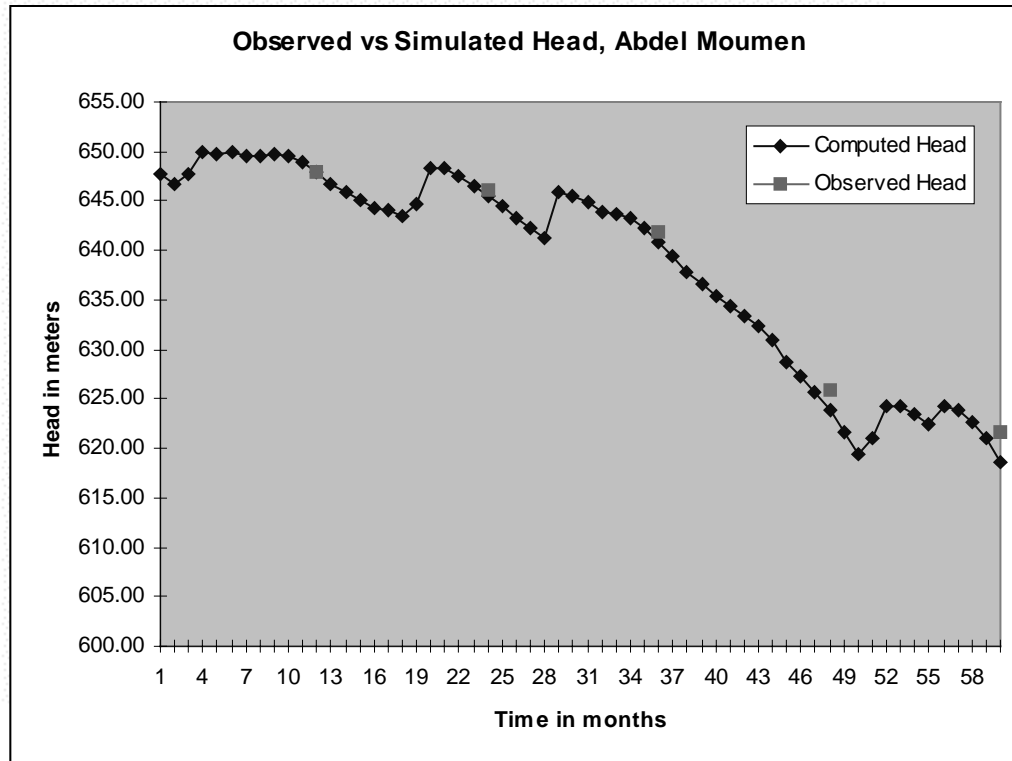


Fig. 5.12: Variation of Reservoir Surface Elevation at Abdel Moumen

Chapter 6: Discussion and Conclusions

6.1 DISCUSSION

The success of the Souss basin simulation effort must be viewed in terms of the extent to which the objectives laid down in chapter 1 have been met. A simulation model was constructed and calibrated from data which are readily available for most major river basins in the world. The process of developing this model was documented from beginning to end. The values of spatially distributed parameters were within reasonable limits. The deviation of observed from simulated flows were also comparable to the targets set for a preliminary planning study. The reservoirs in the basin were successfully incorporated into a map based environment. A mechanism was provided for allowing the user to alter the location of the system components, such as dams, and to observe the changes that resulted. The use of the model as a planning tool was illustrated by demonstrating its use in analyzing some common planning scenarios. Hence, the simulation effort could be said to have been a success in quantitative terms. The model does have its limitations though, and it is important to understand these and the reliability of information it provides in various applications.

The parameter calibration procedure was the most difficult part of the modeling process. The optimization module calibrates model parameters by

performing surface water simulations with various parameters from within the range specified by the user. When it starts to calibrate the first parameter, the values of other parameters which influence flow in the model are still set to zero. Hence initial calibration runs may not obtain parameter values that cause the optimization to converge. It takes several optimization runs to obtain parameter values that generate flows of the order of magnitude observed in the basin. Providing initial estimates of the expected values of parameters during preprocessing would allow the model to converge much faster. Another limitation of the optimization procedure is that when the model is unable to find a value of the parameter that causes the optimization to converge, the last value tested is assigned to the parameter. Other parameters are subsequently tested based on the assumption that a correct value was found for the previous parameter. Again, this can lead to a vicious cycle in which parameters are being calibrated based on flows generated from other incorrect parameters. Default values similar to those proposed for the initial model run should be assigned to parameters when the optimization does not converge. This would allow other parameters to be calibrated based on parameters of a reasonable order of magnitude.

On the temporal scale, the model performed better at a monthly scale than it did at daily scale. The values of the model parameters obtained for the daily model were outside of the expected range of values. The in-stream velocities were

too low while the river loss coefficients were too high. The time series generated by the daily model produced a better quantitative match but it did not bear a close resemblance to the observed daily flows. This may have been due in part to the fact that the model did not undergo as vigorous a calibration to refine the values of the parameters. However, with each calibration run taking up to 12 hours to converge, the scope of this research did not permit investing more effort in calibrating it. An alternative approach would have been to reduce the calibration period from a year to a month, thus reducing the time for each run to converge. However, this approach would have resulted in a biased model as it would only account for conditions in one month of the year. A balance must consequently be struck between the desired accuracy and the calibration effort expended. The model should be tested at a weekly time scale where variations are likely to occur more gradually. This time period may provide the desired balance between processing time and quality of runoff data generated in smaller river basins.

The monthly model on the other hand performed very well during periods of gradually varying flow. Similarity in the shape of the flow series pointed to the fact that the timing of flows was reproduced during such periods. Results were considerably better at the points of calibration and in sections of the river located between two calibration points. This seemed to suggest that the reliability of the model in solving particular problems can be improved considerably if the locations

of interest are identified early in the study and more calibration effort invested in those sections. However, if the model is to be used to answer a variety of questions about various parts of the basin, then spending the required time in calibration of the whole model may be a worthwhile undertaking.

Isolated peaks and troughs were not captured very well by the model. The generated flows take the shape of a wave form with changes in flows being effected gradually over a few time periods. Variations in daily flows tend to occur gradually while monthly flows may sometimes not adhere to this trend. This offers one plausible explanation as to why the daily model produced a better quantitative match than the monthly model. However, the issue of isolated peaks and troughs do not pose a major problem in long term water resource planning. Extreme events such as floods and droughts are most significant in planning when they extend over a long duration. It is during periods of sustained high flow or drought that issues of how much water to store and how much to release become significant at a monthly time scale. For isolated extreme events, flood routing models which operate at much smaller time scales are more suited to keeping track of flows. Such models simulate flow over time steps ranging from a few minutes to a few hours, depending on the size of the basin. They also incorporate hydraulic routing parameters which are not considered in a hydrologic model..

The model also appeared to have an initial startup period covering about two time steps. During this startup period, flows were not simulated. Consequently, the RMSE and SMD did not indicate a close match between observed and simulated flows for time series with a high initial flow. The effect of the initial startup period is less pronounced when the simulation is performed for an extended time period.

The comparison of results from the Niger and Souss basin models suggests that the model performs better in a large basin. This can be attributed to the fact that hydrologic response in small basins is more sensitive to slight changes in response parameters. Smaller basins do not have the damping effect afforded by the large surface area of watersheds in large basins. The effects of variations in land surface elevation are also more pronounced. Consequently, flows tend to vary much more rapidly than in large basins. The model was shown to be sufficient for performing preliminary assessments in small basins. For more detailed design work, the relative effects of spatial variables such as vegetation, land cover, soil and infiltration should be incorporated into the model. This can be done by identifying the model parameters that would be influenced by the variables and assigning relative weights to account for the spatial variations in these variables. Such an approach would perhaps make the model more representative of field conditions by accounting for spatial variations more explicitly.

The surface water model does not update flow on arcs containing reservoirs until the next downstream node. Consequently, flows observed on the section of the river between the reservoir and the next downstream node do not take reservoir discharges into account. This problem arises because flows along river reaches are not stored. They are interpolated, on the fly, from the flows recorded at the upstream ('fflow') and downstream ('tflow') nodes. The flow to the downstream node, the 'tflow', is in turn computed from the upstream flow, 'fflow', and the flow from the watershed polygon, the 'pflow'.

The number of watersheds and river reaches defined in the Souss basin was too high for the amount of runoff data available and the level of accuracy achievable in the reservoir planning phase of the study. This disparity resulted in the expenditure of an inordinately large amount of time and effort during the model calibration phase. No more than three or four watersheds can reasonably be calibrated at a single runoff station. Hence a coverage containing about 20 to 25 watersheds would have been more appropriate to use. Removing some of the minor streams from the river coverage to generate a simpler stream network would have focused more attention on the major tributaries which had the greatest influence on the simulation results. It would also have reduced the time for each calibration run thereby allowing for a more thorough calibration of the entire

model. The stream network also contained a detached stream network which crossed the basin boundary without joining the main river or its tributaries. This stream network and its associated watersheds should have been removed from the basin coverage since it did not contain a runoff station with which to calibrate it.

6.2 CONCLUSIONS

The Souss basin study is the first attempt by anyone other than the original developer, to use GH-Rivers to construct a model of river basin. The construction of the model was completed successfully. The user interface provided for preprocessing and flow simulation activities were easy to use, and the graphical display of simulation processes made it easy for the user to follow the progress of the simulation computation. The parameter calibration portion of the modeling effort was perhaps the weakest link in the chain. The optimization module used in the calibration has a number of limitations which hinder progress in seeking a set of parameters that cause the calibration to converge. This results in an inordinate amount of time spent searching for suitable values of model parameters. The calibration process needs to be reviewed to make it easier for the user to narrow down the range of feasible parameters.

The study of the reservoir planning activities in the Souss basin provided insight into the use of a map based surface water simulation model in water

resource planning. It also provided a basis for developing additional reservoir simulation programs required to make the model into a more effective decision support tool. A user interface for the reservoir simulation programs was developed in the form of an ArcView extension to make them easier to use. This extension called DAMS offers the user a means of interactively changing the configuration of the river model and modifying the water allocations from the reservoirs.

The Souss basin is a complex hydrologic system with many different components that affect the flow and distribution of water in the basin. Models that enable planners to better analyze the performance of the system can only improve the knowledge base for decision making. The hydrologic model was developed based on the best basin data available. In view of ongoing research into ways of minimizing water loss in the basin, it is hoped that the model developed in this study can be incorporated into the decision making process.

This study is only the starting point in the development of GIS based water resource planning tools. There is obviously a lot of room for improving the technology and the processes in map-based, water resource planning. However, the advantages offered by the map-based approach make it a viable option for water resource planning particularly in developing countries which have little resources to invest in the database development associated with river basin modeling.

Appendix 1: Data Dictionary

FILE NAME	DESCRIPTION	LOCATION
raindata.dbf	Uninterpolated rainfall data from 1935 to 1995. Missing records represented as -1.	home/asante/data/rainfall
mrn35.dbf to mrn95.dbf	monthly rainfall data. Each file contains a year of data.	home/asante/data/rainfall/m onthly
rn35.dbf to rn95.dbf	Daily rainfall data. Each file contains a year of data.	home/asante/data/rainfall/da ily
sr0203.dbf sr0594.dbf sr0597.dbf sr0642.dbf sr0858.dbf sr4340.dbf	Daily runoff data. The last four digits are the DGH assigned runoff station number. Missing records represented by -1.	home/asante/data/runoff
d54s0203.dbf to d83s0203.dbf d77s0594.dbf to d95s0594.dbf d79s0597.dbf to d95s0597.dbf d69s0642.dbf to d95s0642.dbf d76s0858.dbf to d95s0858.dbf d73s4340.dfb to d94s4340.dbf	Daily runoff data. Each file contains a year of data. 'd' indicates daily data. The next two digits are the year of record. 's' stands for station. The last four digits are the DGH assigned station number.	home/asante/data/runoff/dai ly
m54s0203.dbf to m83s0203.dbf m77s0594.dbf to m95s0594.dbf m79s0597.dbf to m95s0597.dbf m69s0642.dbf to m95s0642.dbf	Monthly runoff data. Each file contains a year of data. 'm' indicates daily data. The next two digits are the year of record. 's' stands for station. The last four digits are the DGH assigned station number.	home/asante/data/runoff/mo nthly

m76s0858.dbf to m95s0858.dbf m73s4340.dbf to m94s4340.dbf		
nwa_2	The original 30'' DEM of North West Africa.	home/asante/coverage
burn_fil	The processed DEM of the Souss basin with the stream network burned-in.	home/asante/coverage
sobasin	Coverage of the Souss basin watershed. Contains a polygon coverage of the subwatersheds, a line coverage of basin boundaries and a point coverage of label points.	home/asante/coverage
soriver	Line coverage of the Souss basin river network. Contains 114 arc segments.	home/asante/coverage
hydrost	Point coverage of the runoff stations in the Souss basin.	home/asante/coverage
rainst	Point coverage of the rain gage locations in the Souss basin.	home/asante/coverage
bigdams.shp	Shape file of the three main dams in the Souss basin.	home/asante/coverage
aquifer	Coverage of the ground water aquifer located in the middle of the Souss basin.	home/asante/coverage
aoulouz.dbf abdelmen.dbf dhkila.dbf	The original reservoir operation tables containing historical operation data for the reservoirs at Aoulouz, Abdel Moumen and Dhkila, respectively. These files contain the data as transcribed from the original documents.	home/asante/dams
aouldmop.dbf abdmdmop.dbf dkildmop.dbf	Processed reservoir operation tables containing time series data for the reservoirs at Aoulouz, Abdel Moumen and Dhkila, respectively. The data in these files have been	home/asante/dams

	processed to conform to the input data structure of the Souss simulation model.	
aouldmrg.dbf abdmdmrg.dbf dkildmrg.dbf	Reservoir regulation data for defining the Storage-Elevation and Surface Area-Elevation curves. The first four letters of the file names stand for Aoulouz, Abdel Moumen and Dhkila, respectively.	home/asante/dams
dmrunoff.ave	An Avenue program for converting a file containing a year of daily runoff data to monthly values.	home/asante/programs
dmrain.ave	An Avenue program for converting a file containing a year of daily rainfall data to monthly values.	home/asante/programs
cmptotal.ave	An Avenue program for computing total withdrawal/release and updating the values in the reservoir operation tables.	home/asante/programs
setdmval.pre	An Avenue program for updating the values of fields in the reservoir operation tables.	home/asante/programs
setdmrg.pre	An Avenue program for converting reservoir regulation tables into a standardized format readable by the reservoir simulation model.	home/asante/programs
setdmst0.utl	An Avenue program for setting initial reservoir storage at the beginning of a simulation period.	home/asante/programs
cmpsumry.utl	An Avenue program for computing a summary of results at the end of a simulation.	home/asante/programs
insertdm.pre	An Avenue program for activating the Dam locator	home/asante/programs

	tool. Allows dams to be inserted at any location on the stream network in the View window.	
removedm.pre	An Avenue program for removing Dams from the stream network in the View window.	home/asante/programs
Sosprain.c	A C program for reading data stored in two dimensional tables into one dimensional time series	home/asante/programs
Damrt.ave	An Avenue program for computing a monthly reservoir balance. This is a modified version of the original Damrt.ave included in the simulation program.	home/asante/programs
sosprj	A text file containing inputs for projecting coverages from geographic coordinates to the Lambert Conformal Conical projection used in Morocco.	home/asante/programs

Appendix 2: Converting daily to monthly rainfall

PROGRAM DMRAIN

,

This program creates a table of total monthly rainfall

It requires a year of daily rainfall measurements as input

The input files must conform to a file naming convention

Files should have names such as 'rn83.dbf' which

' implies daily rainfall records for the year 1983

The input files may contain data from more than one station

' as long as data from each station is presented in a single column

Station-ids (such as GC1583) are used as field names (or attributes) in the input files

Check to ensure the input table does not contain missing (or negative) records

Add the input files to the project before running the program

TheProject=av.getproject

meanyear = msgbox.input("Enter year to compute e.g. 1985","Input","")

mnyear=(meanyear.asnumber mod 100)

tblname="rn"+mnyear.asstring+".dbf"

outtbl="mrn"+mnyear.asstring+".dbf"

TheTable=TheProject.FindDoc(tblname)

TheVtab=TheTable.GetVtab

theFields=TheVtab.GetFields

'This is for non-leap years

'refer to lpmonavg.ave for leap years

'CREATE A LIST OF MONTH NAMES

'The program assumes the first value is for January 1st

'days in the month for a non-leap year

'Jan (31)

'Feb (28)

'Mar (31)

'Apr (30)

'May (31)

'Jun (30)

'Jul (31)

'Aug (31)

'Sep (30)

'Oct (31)

'Nov (30)

'Dec (31)

day_l=List.Make

day_l.add(31)

day_l.add(59)

day_l.add(90)

day_l.add(120)

day_l.add(151)


```
day_1.add(181)
day_1.add(212)
day_1.add(243)
day_1.add(273)
day_1.add(304)
day_1.add(334)
day_1.add(364)
```

Now make an allowance for leap years

```
if((meanyear.asnumber mod 4)<>0)then
  Ndays=364
else
  Ndays=365
  day_1.set(1,60)
day_1.set(2,91)
day_1.set(3,121)
day_1.set(4,152)
day_1.set(5,182)
day_1.set(6,213)
day_1.set(7,244)
day_1.set(8,274)
day_1.set(9,305)
day_1.set(10,335)
day_1.set(11,365)

end
```

```
`width=thefields.get(5).getwidth
`prec=thefields.get(5).getprecision
`msgbox.info(type.asstring++width.asstring++prec.asstring, "")
```

```
outVtab=Vtab.MakeNew(outtbl.asfilename,dBase)
```

```
oflist=list.make
for each f in thefields
  fname=f.getname
  type=f.gettype
  oflist.add(field.make(fname,type,6,0))
end
```

```
outFlist=oflist.DeepClone
```

```
if(outVtab.CanEdit)then
  outVtab.SetEditable(true)
else
  msgbox.info("Can't edit output table","Exit")
  exit
end
```

```
outVtab.AddFields(outFlist)
outTable=Table.Make(outVtab)
outTable.setname(outVtab.getname)
```

```
`chk1=msgbox.yesno("Do you wish to continue?", "",TRUE)
```

```
'  if(chk1=false)then  
'    exit  
'  end
```

```
outFields=outVtab.GetFields
```

```
datefield=outfields.get(0)
```

```
LOOP THROUGH THE VTAB
```

```
fcount=0
```

```
sum=0
```

```
for each f in theFields
```

```
  if (thefields.find(f)=0) then
```

```
    for each i in 1..12
```

```
      newrec=outVtab.addrecord
```

```
      outvtab.setvalue(datefield,newrec,i)
```

```
    end
```

```
  else
```

```
    count=0
```

```
    mon=0
```

```
    for each rec in 0..Ndays
```

```
      count=count+1
```

```
      prec=thevtab.returnvalue(f,rec)
```

```
    days=day_1.get(mon)
```

```

    if (count>days) then
        outvtab.setvalue(outfields.get(fcount),mon,sum)
        mon = mon+1

        sum=prec
    else
        sum=sum+prec
    end

end 'end of loop through records

sum=0

end 'end of if
fcount=fcount+1
end 'end of field loops
outVtab.SetEditable(false)

```

Appendix 3: Converting daily to monthly runoff

PROGRAM DMRUNOFF

,

This program creates a table of mean monthly flows

It requires a year of daily runoff measurements as input

The input files must conform to a file naming convention

This convention uses filenames such as 'd83s0203.dbf' where

' d83 implies daily flow rates for the year 1983

' s0203 implies surface runoff at station-id = 0203

'Add the input files to the project before running the program

TheProject=av.getproject

The year and runoff station-id are entered

This enables the input file to be identified

Also used to identify leap years

meanyear = msgbox.input("Enter year e.g. 1985", "Input Year", "")

rstation = msgbox.input("Enter runoff station-id e.g. 0203", "Input Station", "")

mnyear=(meanyear.asnumber mod 100)

tblname="d"+mnyear.asstring+"s"+rstation+".dbf"

outtbl="m"+mnyear.asstring+"s"+rstation+".dbf"

TheTable=TheProject.FindDoc(tblname)

TheVtab=TheTable.GetVtab

TheFields=TheVtab.GetFields

'CREATE A LIST OF MONTH NAMES

'The program assumes the first value is for 100 January 1st

'days in the month for a non-leap year

'Jan (31)

'Feb (28)

'Mar (31)

'Apr (30)

'May (31)

'Jun (30)

'Jul (31)

'Aug (31)

'Sep (30)

'Oct (31)

'Nov (30)

'Dec (31)

day_l=List.Make

day_l.add(31)

day_l.add(59)

day_l.add(90)

day_l.add(120)

day_l.add(151)

day_l.add(181)

day_l.add(212)

day_l.add(243)

day_l.add(273)

```
day_1.add(304)
```

```
day_1.add(334)
```

```
day_1.add(365)
```

Now make an allowance for leap years

```
if((meanyear.asnumber mod 4)<>0)then
```

```
  Ndays=365
```

```
else
```

```
  Ndays=366
```

```
  day_1.set(1,60)
```

```
day_1.set(2,91)
```

```
day_1.set(3,121)
```

```
day_1.set(4,152)
```

```
day_1.set(5,182)
```

```
day_1.set(6,213)
```

```
day_1.set(7,244)
```

```
day_1.set(8,274)
```

```
day_1.set(9,305)
```

```
day_1.set(10,335)
```

```
day_1.set(11,366)
```

```
end
```

```
`width=thefields.get(5).getwidth
```

```
`prec=thefields.get(5).getprecision
```

```
`msgbox.info(type.asstring++width.asstring++prec.asstring, "")
```

```

outVtab=Vtab.MakeNew(outtbl.asfilename,dBase)

oflist=list.make
for each f in thefields
    fname=f.getname
    if (fname<>"ndate") then
        type=f.gettype
        oflist.add(field.make(fname,type,8,0))
    end
end

outFlist=oflist.DeepClone

if(outVtab.CanEdit)then
    outVtab.SetEditable(true)
else
    msgbox.info("Can't edit output table","Exit")
    exit
end

outVtab.AddFields(outFlist)
outTable=Table.Make(outVtab)
outTable.setname(outVtab.getname)

'chk1=msgbox.yesno("Do you wish to continue?", "",TRUE)
'    if(chk1=false)then
'        exit
'    end

```



```
outFields=outVtab.GetFields
```

```
datefield=outfields.get(0)
```

```
'LOOP THROUGH THE VTAB
```

```
fcount=0
```

```
sum=0
```

```
for each f in theFields
```

```
  if (thefields.find(f)=0) then
```

```
    for each i in 1..12
```

```
      newrec=outVtab.addrecord
```

```
      outvtab.setvalue(datefield,newrec,i)
```

```
    end
```

```
  else
```

```
    count=0
```

```
    mon=0
```

```
    rdays=0
```

```
    for each rec in 0..Ndays
```

```
      count=count+1
```

```
      prec=thevtab.returnvalue(f,rec)
```

```
      if (prec<0) then
```

```
        prec=0
```

```
        rdays=rdays+1
```

```
      end
```

```

days=day_1.get(mon)

if (count>days) then
  if (mon=0) then
    mrun=sum*24*3600*days/(days-rdays)
  else
    ldays=day_1.get(mon-1)
    thmdays=days-ldays
    mrun=sum*24*3600*thmdays/(thmdays-rdays)
  end
  outvtab.setvalue(outfields.get(fcount),mon,mrun)
  mon = mon+1

  if (mon>11) then
    exit
  end

  sum=prec
else
  sum=sum+prec
end

end 'end of loop through records

sum=0

end 'end of if

```

```
fcount=fcount+1
'chk=msgbox.yesno("continue?", "", true)
if (chk=false) then
' exit
'end
end 'end of field loops

outVtab.SetEditable(false)
```

Appendix 4: Interpolating for Missing Rainfall Records, Cmprain.pre

```
*****
```

```
'Interpolate the Missing Points of rainfall
```

```
'time-series
```

```
*****
```

```
ThmName="RainSt"
```

```
'RainTbName="rn6869.dbf"
```

```
theProject=av.getProject
```

```
theView=av.GetActiveDoc
```

```
TheTheme=TheView.FindTheme(ThmName)
```

```
TblList=list.make
```

```
for each dd in TheProject.GetDocs
```

```
  if(dd.GetGUI="table")then
```

```
    tblList.add(dd.getname)
```

```
  end
```

```
end
```

```
raintbname=msgbox.choiceAsString(tblList,"SelectRainfallTimeSeriesTable","RainTimeSeries")
```

```
if (raintbname=nil) then
```

```
  exit
```

```
end
```

```

TheFtab=TheTheme.GetFtab
Gcnt=TheView.GetGraphics.Count
ShpFld=TheFtab.FindField("Shape")
IdFld=TheFtab.FindField("RainId")
Trec=theFtab.GetNumRecords
DistDict=Dictionary.Make(Trec)
IDDict=Dictionary.Make(Trec)
RainTb=TheProject.Finddoc(raintbname)
RainVtab=rainTb.GetVtab
  If(RainVtab.CanEdit)then
    RainVtab.SetEditable(true)
  end
Tmfld=RainVtab.FindField("Ndate")
av.ShowStopButton
Cancelled=False
for each rec in TheFtab
  shpV=TheFtab.ReturnValue(ShpFld,rec)
  shpVX=shpV.GetX
  shpVY=shpV.GetY
  pntIDV=TheFtab.ReturnValue(IdFld,rec)
  rainFld=RainVtab.FindField("GC"+PntIdV.AsString)
  Gpnt = GraphicShape.Make(ShpV)
  Asym = Gpnt.GetSymbol
  Asym.SetColor(Color.getGreen)
  theView.GetGraphics.Add(Gpnt)
  av.showmsg("Processing station:;++PntIDV.AsString)
  Cancelled=av.SetStatus(rec/Trec*100).not

```

```

if(Cancelled)then
    *****

    'Remove Graphics
    *****

    MGlist=TheView.GetGraphics
    GFcnt=MGlist.count-1
    for each g in Gcnt..GFcnt
        MGList.Get(g).SetSelected(true)
    end
    MGlist.ClearSelected
    *****

    'Remove Graphics
    *****

    exit
end
*****

'setup dictionary
*****

DistDict.Empty
IDDict.Empty
for each lrec in TheFtab
    if(lrec=rec)then
        continue
    end
    TheIdV=TheFtab.ReturnValue(IdFld,lrec)
    NpntV=TheFtab.ReturnValue(ShpFld,lrec)
    NpntX=NpntV.GetX
    NpntY=NpntV.GetY

```

```

    DistV=((NpntX-shpVX)^2+((NpntY-shpVY)^2))
    DistDict.Add(TheIdV,DistV)
    IDDict.add(TheIDV,"GC"+TheIDV.AsString)
end
for each tmrec in rainVtab
    TheRain=RainVtab.ReturnValue(rainFld,tmrec)
    if(theRain < 0.0)then
        ThrRain=0.0
        TheSum=0.0
        Asum=0.0
        for each llrec in TheFtab
            if(llrec=rec)then
                continue
            end
            TheIdV=TheFtab.ReturnValue(IdFld,llrec)
            nextrain=RainVtab.Findfield(IDDict.get(TheIDV))
            tmpV=RainVtab.ReturnValue(nextrain,tmrec)
            if(tmpV > -0.11)then
                TheDist=DistDict.Get(TheIdV)
                DistV=((TheDist)^(-1))
                TheSum=TheSum+(tmpV*DistV)
                Asum=Asum+DistV
            end
        end
        for each llrec in TheFtab
            if(Asum>0)then
                tmpV=TheSum/Asum
            else
                tmpV=0.0
            end
        end
    end
end

```

```

        end
        RainVtab.SetValue(RainFld,tmrec,tmpV)
    end if(theRain=-1)
end for each tmrec in rainVtab

end

msgbox.info("Computation completed","EXIT")

*****

Remove Graphics
*****

MGlist=TheView.GetGraphics
GFcnt=MGlist.count-1
for each g in Gcnt..GFcnt
    MGList.Get(g).SetSelected(true)
end
MGlist.ClearSelected
*****

Remove Graphics
*****

"/home/ye/soflow/dosave/cmprain.pre of 105 lines,
"Written on 1996:07:05-05:07:49 by YEZ.

"/home/ye/morocco/sflow/dosave/cmprain.pre of 116 lines,
"Written on 1996:08:20-03:53:26 by YEZ.

```


Appendix 5: Computing Surplus Runoff, Cmpsurp.pre

```
*****
```

```
`Compute surplus for each subwatershed
```

```
`from rainfall data
```

```
,
```

```
*****
```

```
kk=0
```

```
`RainTbName="rn6869.dbf"
```

```
theProject=av.getProject
```

```
theView=av.GetActiveDoc
```

```
ThemeList=TheView.Getthemes
```

```
  pntList=list.make
```

```
  PolyList=list.make
```

```
  for each thm in Themelist
```

```
    if(thm.getFtab.FindField("Shape").getType=#FIELD_SHAPEPOINT)then
```

```
      PntList.add(thm)
```

```
    end
```

```
    if(thm.getFtab.FindField("Shape").getType=#FIELD_SHAPEPOLY)then
```

```
      Polylist.add(thm)
```

```
    end
```

```
  end
```

```
TblList=list.make
```

```
for each dd in TheProject.GetDocs
```

```

if(dd.GetGUI="table")then
  tbllist.add(dd.getname)
end
end

```

```

tblName=msgbox.choiceAsString(tbllist,"SelectRainfallTimeSeriesTable","RainTimeSeries")
if (tblName=nil) then
  exit
end

```

```

RainTheme=msgbox.choiceAsString(pntlist,"RainStationTheme","Rain-Station")
if (RainTheme=nil) then
  exit
end

```

```

PolyTheme=msgbox.choiceAsString(Polylist,"EnterWatershedThemeToInterpolateRainfall","PolyTheme")
if (PolyTheme=nil) then
  exit
end

```

```

RainFtab=RainTheme.GetFtab
PolyFtab=PolyTheme.GetFtab

```

```

PolyIdFld=msgbox.choiceAsString(PolyFtab.GetFields,"Enter Watershed's KeyField"+nl+"eg.Grid-Code",
"TheFieldUsedToConcatenateDataTable'sFieldHeader")
if (PolyIdFld=nil) then

```

```
    exit
end
```

```
RainIdFLd=msgbox.choiceAsString(RainFtab.GetFields,"Enter RainStation's  
KeyField","FieldUsedToConcatenateRainTable'sFieldHeader")
```

```
    if (RainIdFLd=nil) then  
        exit  
    end
```

```
Gcnt=TheView.GetGraphics.Count  
RainShp=RainFtab.FindField("Shape")  
PolyShp=PolyFtab.FindField("Shape")  
RainIdFLd=RainFtab.FindField("RainId")  
PolyIdFLd=PolyFtab.FindField(PGCNAME)  
Trec=PolyFtab.GetNumRecords  
DistDict=Dictionary.Make(Trec)  
IDDict=Dictionary.Make(Trec)  
RainTb=TheProject.FindDoc(tblName) 'you need to enter a rainfall.dbf name here  
RainVtab=rainTb.GetVtab  
    If(RainVtab.CanEdit)then  
        RainVtab.SetEditable(true)  
    end  
surpTb=TheProject.FindDoc("Psurp.dbf")  
SurpVtab=SurpTb.GetVtab  
    if(SurpVtab.canEdit)then  
        SurpVtab.SetEditable(true)  
    end
```

```

TmFld=RainVtab.FindField("Ndate")
av.ShowStopButton
Cancelled=False
SumtmpV=0.0
for each rec in PolyFtab
    kk=kk+1
    shpV=PolyFtab.ReturnValue(PolyShp,rec).ReturnCenter
    shpVX=shpV.GetX
    shpVY=shpV.GetY
    PlyIDV=PolyFtab.ReturnValue(PolyIdFld,rec)
    SurpFld=SurpVtab.FindField("GC"+PlyIDV.AsString)
    Gpnt = GraphicShape.Make(ShpV)
    Asym = Gpnt.GetSymbol
    Asym.SetColor(Color.getRed)
    theView.GetGraphics.Add(Gpnt)
    av.showmsg("Processing
watershed:"++plyIDV.AsString++"lastSurp(CyanPnt)="+SumtmpV.AsString++"
mm/yr")
    Cancelled=av.SetStatus(rec/Trec*100).not
    if(Cancelled)then
        *****
        Remove Graphics
        *****
        MGlist=TheView.GetGraphics
        GFcnt=MGlist.count-1
        for each g in Gcnt..GFcnt
            MGList.Get(g).SetSelected(true)
        end

```

```

MGlist.ClearSelected

*****

Remove Graphics

*****

exit

end

*****

'setup dictionary

*****

DistDict.Empty
IDDict.Empty
for each lrec in RainFtab
    TheIdV=RainFtab.ReturnValue(RainIdFld,lrec)
    NpntV=RainFtab.ReturnValue(RainShp,lrec)
    NpntX=NpntV.GetX
    NpntY=NpntV.GetY
    DistV=((NpntX-shpVX)^2+((NpntY-shpVY)^2))
    DistDict.Add(TheIdV,DistV)
    IDDict.add(TheIDV,"GC"+TheIDV.AsString)
end
SumTmpV=0.0
TmNrec=RainVtab.GetNumRecords
, tmNrec=12
for each tmrec in 0..(TmNrec-1)
    ThrRain=0.0
    TheSum=0.0
    Asum=0.0
    for each llrec in RainFtab

```

```

TheIdV=RainFtab.ReturnValue(RainIdFld,llrec)
nextrain=RainVtab.Findfield(IDDict.get(TheIDV))
tmpV=RainVtab.ReturnValue(nextrain,tmrec)
  TheDist=DistDict.Get(TheIdV)
  DistV=((TheDist)^(-1))
  TheSum=TheSum+(tmpV*DistV)
  ASum=ASum+DistV
end 'for each llrec in RainFtab
if(ASum>0)then
  tmpV=TheSum/ASum
else
  tmpV=0.0
end
*****

'change this section to a percentage as needed
'e.g. tmpV=tmpV*0.25
*****

, tmpV=tmpV*36.5/12
,   if(tmpV < 35.0/12)then
,     tmpV=0.0
,   elseif(tmpV > 395.5/12)then
,     tmpV=tmpV*0.95/30.4
,   else
,     tmpV=0.002108*(2.71828^(0.0307*tmpV))/30.4
,   end
*****

'end of change this section to a percentage as needed
'rainfall in 1/10millimeters

```

```

*****

tmpV=tmpV/3650
SumTmpV=SumTmpV+tmpV
SurpVtab.SetValue(SurpFld,tmrec,tmpV)
end  'for each tmrec in SurpVtab
    Asym.SetColor(Color.getCyan)
    theView.GetGraphics.Add(Gpnt)
    if(kk=1)then
        OldGpnt=Gpnt
    end
    if(kk>=2)then
        oldoldGpnt=OldGpnt
        OldGpnt=Gpnt
        Bsym=oldoldGpnt.GetSymbol
        Bsym.SetColor(Color.getGreen)
        theView.GetGraphics.Add(OldOldGpnt)
    end
end  'for each rec in PloyFtab
*****

'Remove Graphics
*****

msgbox.info("Computation completed","EXIT")
MGlist=TheView.GetGraphics
GFcnt=MGlist.count-1
for each g in Gcnt..GFcnt
    MGList.Get(g).SetSelected(true)
end
MGlist.ClearSelected

```

Remove Graphics

"/home/ye/soflow/dosave/cmprain.pre of 105 lines,

"Written on 1996:07:05-05:07:49 by YEZ.

"/home/ye/morocco/sflow/dosave/cmrsurp.pre of 118 lines,

"Written on 1996:08:19-02:30:51 by YEZ.

"/home/ye/morocco/sflow/dosave/cmrsurp.pre of 143 lines,

"Written on 1996:08:20-04:39:14 by YEZ.

Appendix 6: Script for Inserting Dams, Insertdm.ave

This program goes directly to the insert dams mode
It bypasses the initial dialog used by the PennantRed tool
TheProject=av.getProject

These two lines are required in the event that
the tool I want to use is already selected
reset = av.GetActiveGUI.GetToolBar.GetControls.Get(0)
reset.select

Now, I select the tool I want to use
myTool = av.GetActiveGUI.GetToolBar.GetControls.Get(2)
mytool.select

Appendix 7: Script for Removing Dams, Removedm.ave

'Asks if user wants to start a new set of points

'Set the coverage name variables

TheProject=av.getProject

CtrlList=TheProject.GetObjectTag

if((CtrlList=nil) or (CtrlList.Count<>3))then

 msgbox.info("CtrlList is

Incorrect"+nl+"ResetTheListByClickOnTheStarBUTTON", "ClickStarButtonFrom
ViewButtlnBar")

 exit

end

RivList=CtrlList.get(1).asTokens("=")

RthmName=RivList.get(0)

RGCName=RivList.get(2)

chkpoint="flowchk.shp"

dampoint="Dams.Shp"

TheView=av.GetActiveDoc

```
chkdelete=msgbox.YESNO("Do you want to remove the selected  
Dam(s)?","Remove Dam(s)?",true)
```

```
if(chkdelete=true)then
```

```
  'chk0="2"
```

```
  'SELF.SetTag(chk0)
```

```
DamTheme=TheView.FindTheme(DamPoint)
```

```
  if(DamTheme=nil)then
```

```
    msgbox.error(DamPoint+"IsNotFound,AddtheThemeBeforeContinue","Select.Shp  
    ")
```

```
      exit
```

```
    end
```

```
TheTheme=TheView.FindTheme(RthmName)
```

```
  if(TheTheme=nil)then
```

```
    msgbox.error(RthmName++"IsNotFound,AddtheThemeBeforeContinue",RthmNa  
    me)
```

```
      exit
```

```
    end
```

```
DamFtab=DamTheme.getFtab
```

```
  if(DamFtab.CanEdit)then
```

```
    DamFtab.SetEditable(True)
```

```
else
  msgbox.info(DamFtab.Getname+"CanEdit",DamFtab.CanEdit.AsString)
exit
end
```

```
RFtab=TheTheme.getFtab
```

```
if(RFtab.CanEdit)then
  RFtab.SetEditable(True)
else
  msgbox.info(RFtab.Getname+"CanEdit",RFtab.CanEdit.AsString)
exit
end
```

```
thebitmap=DamFtab.Getselection
if(thebitmap=nil)then
  msgbox.info("No dams selected"+nl+"None deleted", "")
exit
end
```

```
numrecs=0
rivgclst=list.make
damidlst=list.make
thedocs=list.make
```

```
gcfld=Damftab.findfield("Grid_Code")
damidfld=Damftab.findfield("Damid")
```

```

recordlst=list.make

for each rec in thebitmap
thegcode=Damftab.ReturnValue(gcfld,rec)
thedamid=Damftab.ReturnValue(damidfld,rec)
rivgclst.add(thegcode.clone)
damidlst.add(thedamid.clone)
recordlst.add(rec.clone)

numrecs=numrecs+1
end

'msgbox.ListAsString( recordlst, "", "BITMAP")

if(recordlst.isempty)then
msgbox.info("No Dams Selected"+nl+"No Dams Deleted","")
exit
end

if(rivgclst.isempty)then
msgbox.info("Grid_Code Field not found in Dams.dbf"+nl+"No Dams
Deleted","")
exit
end

if(damidlst.isempty)then
msgbox.info("Damid Field not found in Dams.dbf"+nl+"No Dams Deleted","")
exit
end

```

```

thebitmap.clearall
DamFtab.updateSelection
damtheme.clearselection

for each rec in recordlst
damFtab.removeverecord(rec)
end

for each rec in 0..(numrecs-1)
tabid=damidlst.get(rec)
damtab="dam"+tabid.asstring+".dbf"
mydoc=theproject.FindDoc(damtab)
thedocs.add(mydoc)
end

if (theDocs.IsEmpty) then
return nil
end

doc_names = ""
all = true
result = true

if(thedocs isempty)then

for each d in theDocs
' if (NOT all) then

```

```

' result = MsgBox.AllYesNo( "Are you sure you want to delete"++d.GetName,
"Delete", NIL )
' if (result = nil) then
'   all = true
' elseif (result = false) then
'   continue
' end
' end
if(d.Is(View)) then
  editThm = d.GetEditableTheme
  if (editThm <> nil) then
    doSave = MsgBox.YesNoCancel("Save Edits to "+editThm.GetName+" in "+
d.GetName+"?", "Stop Editing", true)
    if (doSave = nil) then
      continue
    end
    if (editThm.StopEditing(doSave).Not) then
      MsgBox.Info("Unable to Save Edits to Theme "
+ editThm.GetName +
", please use the Save Edits As option", "")
      continue
    else
      d.SetEditableTheme(NIL)
    end
  end
end
end
if (d.Is(Table)) then
  if (d.GetVTab.IsBeingEditedWithrecovery) then

```

```

doSave = MsgBox.YesNoCancel("Save Edits to the table "+d.GetName+
"?", "Stop Editing", True)
if (doSave = nil) then
    continue
end
if (d.GetVtab.StopEditingWithRecovery(doSave).Not) then
    MsgBox.Info("Unable to Save Edits to Table " + d.GetName +
        ", please use the Save Edits As option", "")
    continue
end
end
end
if(thedocs.isempty.not)then
for each d in thedocs
av.GetProject.RemoveDoc( d )
end
end
end 'for each d in the docs

end 'if(thedocs.isempty.not)
HasDamFld=RFtab.FindField("HasDam")
grdcodfld=RFtab.FindField("Grid_Code")

rivBitMap=RFtab.GetSelection
rivBitMap.clearall

for each mrec in 0..(numrecs-1)

```



```

rivgcode=rivgclst.get(mrec)
expr="([Grid_Code] = "+rivgcode.asstring+" )"
Rftab.Query(expr,rivBitMap,#VTAB_SELTYPE_OR)
Rftab.UpdateSelection
end

for each rec in rivbitmap
Rftab.SetValue(hasdamfld,rec,0)
end
rivBitMap.clearall
Rftab.updateselection

damtheme.setvisible(false)
damTheme.setActive(true)
chkdelete=false
end 'end if(chkdelete = true) for removing dams

```

Appendix 8: Script for Making Water Allocations, Setdmval.ave

This program updates the values of records in the dam(id).dbf tables

It enables reservoir operation data to be read into model tables from other database tables

Inputs include dams.dbf and a table containing reservoir operation data

```
theproject=av.getproject
```

```
theTable=theProject.finddoc("dams.dbf")
```

```
TheList=av.GetProject.getdocs
```

```
namelist=list.make
```

```
for each dd in thelist
```

```
  if(dd.GetGUI="table")then
```

```
    namelist.add(dd.getname)
```

```
  end
```

```
end
```

```

theVtab=theTable.GetVtab

if(theVtab.IsEditable)then

    theVtab.seteditable(true)

end

damlst=list.make

for each rec in theVtab

    dmid=theVtab.FindField("damid")

    damIdNum=theVtab.returnvalue(dmid,rec)

    damlst.add(damIdNum)

end

DamNum=msgbox.choiceAsString(damlst,"Select A Dam","Dam To Update")

if(damNum=nil)then

    msgbox.info("No Dam Selected","")

    exit

end

mydam=theProject.finddoc("Dam"+damNum.asString+".dbf")

if(myDam=nil)then

```

```

msgbox.info("Add Dam"+DamNum.asstring+".dbf to the project","")

exit

end

thedam=myDam.GetVtab

if(theDam=nil)then

msgbox.info("Dam"+DamNum.asstring+".dbf"+"Not Found","")

exit

end

    theDam.seteditable(true)

damrecs=theDam.GetNumRecords

numyrs=damrecs/12

mylag=((damrecs)mod(12))

if(mylag<>0)then

numyrs=(numyrs).ceiling

else

mylag=12

end

```

```

    Sourcetbl=msgbox.choiceAsString(namelist,"Select table containing
reservoir"+nl+" operation data for dam"+damnum.asstring,"Source Table")

    if(sourcetbl=nil)then

        exit

    end

    srctbl=theproject.finddoc(sourcetbl.asstring)

    if(srctbl=nil)then

        msgbox.info("Add source table "+sourcetbl.asstring+"to the project","")

        exit

    end

    srcvtab=srctbl.getvtab

    if(srctbl=nil)then

        msgbox.info("Source table "+sourcetbl.asstring+" not found","")

        exit

    end

    m=0

    While(m<>99)

        srcfields=srcvtab.getfields

```

```
Sourcefld=msgbox.choiceAsString(srcfields,"Select a source field containing  
"+nl+" input data for "+thedam.asstring,"Source Field")
```

```
if(sourcefld=nil)then
```

```
exit
```

```
end
```

```
dmdfldlst=thedam.getfields
```

```
Targetfld=msgbox.choiceAsString(dmdfldlst,"Select a target field in  
"+thedam.asstring,"Targetfld")
```

```
if(targetfld=nil)then
```

```
msgbox.info("No target field selected","")
```

```
exit
```

```
end
```

```
flowlist=list.make
```

```
srcrecs=srcvtab.getnumrecords
```

```
for each rec in 0..(srcrecs-1)
```

```
myvalue=srcvtab.returnvalue(sourcefld,rec)
```

```
flowlist.add(myvalue)
```

```
end
```

```

if(flowlist.isEmpty)then

    msgbox.info("No records found in source field","")

    exit

end

for each year in 0..(numyrs-1)

    drec=year*12

    lagmon=11

    if(year=(numyrs-1))then

        lagmon=mylag-1

    end

    'msgbox.info("Lagmon = "+lagmon.asstring,"")

    av.ShowStopButton

    av.ShowMsg("Updating Values...")

    for each month in 0..Lagmon

        drec=drec+1

        thedam.SetValue(targetfld,drec-1,flowlist.get(month))

    end

end

```

```
check=msgbox.YESNO("Do you want to update values "+nl+"for more fields in  
"+thedam.asstring+" ?", "check.ave", false)
```

```
    if(check=false)then
```

```
        m=99
```

```
    end
```

```
end
```


Appendix 9: Script for Setting Up Regulation Tables, Setdmrg.ave

This program creates the reservoir regulation tables

These tables contain reservoir Storage - Surface Area - Elevation characteristic data

The inputs to the program include the dams.dbf, an input table containing the reservoir characteristic data

The input table must have a key field, as well as storage, area and elevation fields

```
*****  
*****
```

```
theproject=av.getproject
```

```
theTable=theProject.finddoc("dams.dbf")
```

```
theVtab=theTable.GetVtab
```

```
if(theVtab.IsEditable)then
```

```
    theVtab.seteditable(true)
```

```
end
```

```
damlst=list.make
```

```

for each rec in theVtab

    dmid=theVtab.FindField("damid")

    damIdNum=theVtab.returnvalue(dmid,rec)

    damlst.add(damIdNum)

end

if(damlst.isempty)then

    msgbox.info("No Dams Inserted in the Model","")

    exit

end

DamNum=msgbox.choiceAsString(damlst,"Select Reservoir to Setup Regulation
Data","DamToSetup")

if(damNum=nil)then

    msgbox.info("No Dam Selected","")

    exit

end

TheList=av.GetProject.getdocs

namelist=list.make

for each dd in thelist

```

```

if(dd.GetGUI="table")then

    namelist.add(dd.getname)

end

end

TblName=msgbox.choiceAsString(namelist,"Pick the table containing regulation
data for dam"+damnum.asstring+"","Res-Reg-Tbl ")

if(TblName=nil)then

    exit

end

resTable=av.GetProject.FindDoc(TblName)

resVtab=resTable.GetVtab

resFields=resVtab.GetFields

' if(theVtab.CanEdit)then

'   TheVtab.SetEditable(true)

' end

indexName=msgbox.choiceAsString(resfields,"Select the key field"+nl+"eg.
Index","Key Field")

if(indexName=nil)then

```

```

exit

end

headName=msgbox.choiceAsString(resfields,"Select the reservoir surface
elevation field"+nl+"eg. Head","Elevation in m")

if(headname=nil)then

exit

end

areaName=msgbox.choiceAsString(resfields,"Select the reservoir surface area
field"+nl+"eg. Area","Surface Area in m^2")

if(areaName=nil)then

exit

end

storName=msgbox.choiceAsString(resfields,"Select the reservoir storage
field"+nl+"eg. Storage","Storage in m^3")

if(storName=nil)then

exit

end

rindexfld=resVtab.FindField(indexname.asstring)

rheadfld=resVtab.FindField(headname.asstring)

```

```

rareafld=resVtab.FindField(areaname.asstring)

rstorfld=resVtab.FindField(storname.asstring)

thetbl=theProject.finddoc("dmrg"+damnum.asstring+".dbf")

    if(thetbl<>nil)then

        chkdel=msgbox.YesNo("Delete existing "+thetbl.asstring+" ?", "Delete
table?",true)

        if(chkdel=true)then

            theproject.removedoc(thetbl)

        end

        if(chkdel=false)then

            exit

        end

    end

    if(thetbl=nil)then

        thename="summary.dbf"

        outtbl1=Vtab.MakeNew(theName.asfilename,dbase)

        theTbl=Table.Make(outtbl1)

        thetbl.setname("dmrg"+damnum.asstring+".dbf")

```

```

av.GetProject.AddDoc(theTbl)

Nfldlst=list.make

Nfldlst.Add(field.Make("Index",#field_decimal,5,0))

Nfldlst.Add(field.Make("Head",#field_decimal,5,0))

Nfldlst.Add(field.Make("Area",#field_decimal,16,0))

Nfldlst.Add(field.Make("Storage",#field_decimal,16,0))

outlist=Nfldlst.DeepClone

outtbl1.AddFields(outlist)

'end 'end of if demand.dbf not found

outtbl=thetbl.GetVtab

outtbl.SetEditable(true)

rectoadd=resVtab.getnumrecords

indexfld=outtbl.FindField("Index")

headfld=outtbl.FindField("Head")

areafld=outtbl.FindField("Area")

storfld=outtbl.FindField("Storage")

for each rec in 0..(rectoadd-1)

rec=outtbl.AddRecord

```

```
rindexval=resVtab.ReturnValue(rindexfld,rec)

rheadval=resVtab.ReturnValue(rheadfld,rec)

rareaval=resVtab.ReturnValue(rareafld,rec)

rstorval=resVtab.ReturnValue(rstorfld,rec)

outtbl.setvalue(indexfld,rec,rindexval)

outtbl.setvalue(headfld,rec,rheadval)

outtbl.setvalue(areafld,rec,rareaval)

outtbl.setvalue(storfld,rec,rstorval)

end
```

Appendix 10: Script for Initializing Reservoir Storage, Setdmst0.ave

'This program initializes reservoir storage

'Run it from the dams menu to reset the field storage0

'Inputs are dams.dbf and the initial storage

theproject=av.getproject

theTable=theProject.finddoc("dams.dbf")

theVtab=theTable.GetVtab

if(theVtab.IsEditable)then

 theVtab.seteditable(true)

end

damlst=list.make

storfld=theVtab.findfield("Storage0")

damidfld=theVtab.findfield("damid")

for each rec in theVtab


```

damIdNum=theVtab.returnvalue(damidfld,rec)

damlst.add(damIdNum)

end

m=0

While(m<>99)

DamNum=msgbox.choiceAsString(damlst,"Select A Dam to
initialize","DamToAlter")

if(damNum=nil)then

msgbox.info("No Dam Selected","")

exit

end

theBitMap=theVtab.GetSelection

theBitMap.Clearall

expr="([Damid] = "+damnum.asstring+" )"

theVtab.Query(expr,theBitMap,#VTAB_SELTYPE_NEW)

theVtab.UpdateSelection

inistore=msgbox.Input("Enter initial storage in m3","Set
Storage0",1000000.asstring)

if(inistore=nil)then

```

```

exit

end

    for each rec in thebitmap

        theVtab.SetValue(storfld,rec,inistore)

    end

    theBitMap.clearall

    TheVtab.updateselection

    check=msgbox.YESNO("Do you want to initialize another
dam?", "check.ave", false)

    if (check=false) then

        m=99

    end

end

theVtab.seteditable(false)

```

Appendix 11: Script for Summing Up Monthly Outflows, Cmptotal.ave

'This program computes total withdrawal/release

'when there are multiple values within each time period

'The user is prompted on whether to update the dam(id).dbf tables

'Inputs include dams.dbf and a table containing reservoir operation data

theproject=av.getproject

theTable=theProject.finddoc("dams.dbf")

TheList=av.GetProject.getdocs

namelist=list.make

for each dd in thelist

if(dd.GetGUI="table")then

namelist.add(dd.getname)

end

```

end

theVtab=theTable.GetVtab

if(theVtab.IsEditable)then

    theVtab.seteditable(true)

end

damlst=list.make

for each rec in theVtab

    dmid=theVtab.FindField("damid")

    damIdNum=theVtab.returnvalue(dmid,rec)

    damlst.add(damIdNum)

end

DamNum=msgbox.choiceAsString(damlst,"Select A Dam","DamToAlter")

if(damNum=nil)then

    msgbox.info("No Dam Selected","")

exit

end

mydam=theProject.finddoc("Dam"+damNum.asString+".dbf")

if(myDam=nil)then

```

```

msgbox.info("Dam"+DamNum.asstring+".dbf"++"Not Found","")

exit

end

thedam=myDam.GetVtab

theDam.seteditable(true)

damrecs=theDam.GetNumRecords

numyrs=damrecs/12

mylag=((damrecs)mod(12))

if(mylag<>0)then

numyrs=(numyrs).ceiling

else

mylag=12

end

while(true)

cmpdmd=msgbox.input("2=Compute Withdrawal"+nl+"1=Compute Release"

+nl+"0=Cancel","Your Choice...","2")

if((cmpdmd<>"0") and (cmpdmd<>"1") and (cmpdmd<>"2"))then

    chk0=Msgbox.YesNo("Enter: 0, 1, or 2 to select","Continue...?",true)

```

```

    if(chk0)then

        continue

    else

        exit

    end

else

break

end

end

if(cmpdmd="2")then

    thename0="Demand"+damnum.asstring+".dbf"

    thetbl=theProject.finddoc(thename0)

    if(thetbl<>nil)then

        theproject.removedoc(thetbl)

    end

    outtbl1=Vtab.MakeNew(theName0.asfilename,dbase)

    theTbl=Table.Make(outtbl1)

    thetbl.setname(thename0)

```

```

av.GetProject.AddDoc(theTbl)

Nfldlst=list.make

Nfldlst.Add(field.Make("Irrigation",#field_decimal,10,0))

Nfldlst.Add(field.Make("Industrial",#field_decimal,10,0))

Nfldlst.Add(field.Make("WaterSupply",#field_decimal,10,0))

Nfldlst.Add(field.Make("Other",#field_decimal,10,0))

Nfldlst.Add(field.Make("Withdrawal",#field_decimal,10,0))

outlist=Nfldlst.DeepClone

outtbl1.AddFields(outlist)

outtbl=thetbl.GetVtab

outtbl.SetEditable(true)

irrfld=outtbl.FindField("Irrigation")

indfld=outtbl.FindField("Industrial")

wsfld=outtbl.FindField("WaterSupply")

otherfld=outtbl.FindField("Other")

wthdrfld=outtbl.FindField("Withdrawal")

theTbl.GetWin.Open

  for each recs in 1..damrecs

```

```

rec=outtbl.AddRecord

outtbl.SetValue(irrifld,rec,0)

outtbl.SetValue(indfld,rec,0)

outtbl.SetValue(wsfld,rec,0)

outtbl.SetValue(otherfld,rec,0)

outtbl.SetValue(wthdrfld,rec,0)

end

outtbl.SetEditable(true)

dmdfldlst={ irrifld,indfld,wsfld,otherfld,wthdrfld }

Sourcetbl=msgbox.choiceAsString(namelist,"Select a source table
containing"+nl+" withdrawal data","Source Table")

if(sourcetbl=nil)then

exit

end

srctbl=theproject.finddoc(sourcetbl.asstring)

if(srctbl=nil)then

msgbox.info("Add source table "+sourcetbl.asstring+"to the project","")

exit

```



```

end

srcvtab=srctbl.getvtab

if(srctbl=nil)then

msgbox.info("Source table "+sourcetbl.asstring+"not found","")

exit

end

m=0

While(m<>99)

srcfields=srcvtab.getfields

Sourcefld=msgbox.choiceAsString(srcfields,"Select a source field containing
a"+nl+" component of total withdrawal","Source Field")

if(sourcefld=nil)then

exit

end

Targetfld=msgbox.choiceAsString(dmdfldlst,"Select a target field in
demand.dbf","Targetfld")

if(targetfld=nil)then

msgbox.info("No target field selected","")

exit

```

```

end

flowlist=list.make

srcrecs=srcvtab.getnumrecords

for each rec in 0..(srcrecs-1)

myvalue=srcvtab.returnvalue(sourcefld,rec)

flowlist.add(myvalue)

end

if(flowlist.isEmpty)then

msgbox.info("No records found in source field","")

exit

end

for each year in 0..(numyrs-1)

drec=year*12

lagmon=11

if(year=(numyrs-1))then

lagmon=mylag-1

end

msgbox.info("Lagmon = "+lagmon.asstring,"")

```

```

for each month in 0..lagmon

    drec=drec+1

    outtbl.SetValue(targetfld,drec-1,flowlist.get(month))

    thetbl.GetWin.Invalidate

end

end

check=msgbox.YESNO("Do you want to input data "+nl+"for more withdrawal
fields?","check.ave",false)

if(check=false)then

    m=99

end

end

for each srec in 1..damrecs

    irridmd=outtbl.ReturnValue(irrifld,srec-1)

    inddmd=outtbl.ReturnValue(indfld,srec-1)

    wsdmd=outtbl.ReturnValue(wsfld,srec-1)

    otherdmd=outtbl.ReturnValue(otherfld,srec-1)

    withdraw=irridmd + inddmd+ wsdmd + otherdmd

```

```

        outtbl.SetValue(wthdrfld,srec-1,withdraw)

    end

    thetbl.GetWin.Invalidate

    chkupdate=msgbox.YESNO("Do you want to update withdrawal in
    "+thedam.asstring+" now?", "Update Withdrawal?",true)

    damwthdr=theDam.FindField("Withdraw")

    if(chkupdate=true)then

        av.ShowStopButton

        av.ShowMsg("Updating Withdrawal...")

        for each srec in 1..damrecs

            updwthdr=outtbl.ReturnValue(wthdrfld,srec-1)

            thedam.SetValue(damwthdr,srec-1,updwthdr)

        end

    end

    av.ClearStatus

end 'end of yes compute withdrawal

if(cmpdmd="1")then

    thename="release"+damnum.asstring+".dbf"

```

```

    thetbl=theProject.finddoc(thename.asstring)

    if(thetbl<>nil)then

    theproject.removedoc(thetbl)

end   outtbl1=Vtab.MakeNew(theName.asfilename,dbase)

    theTbl=Table.Make(outtbl1)

    thetbl.setname(thename)

    av.GetProject.AddDoc(theTbl)

    Nfldlst=list.make

    Nfldlst.Add(field.Make("Release1",#field_decimal,16,0))

    Nfldlst.Add(field.Make("Release2",#field_decimal,16,0))

    Nfldlst.Add(field.Make("Release3",#field_decimal,16,0))

    Nfldlst.Add(field.Make("Release4",#field_decimal,16,0))

    Nfldlst.Add(field.Make("Total",#field_decimal,16,0))

    outlist=Nfldlst.DeepClone

    outtbl1.AddFields(outlist)

    outtbl=thetbl.GetVtab

    outtbl.SetEditable(true)

```

```

irrifld=outtbl.FindField("Release1")

indfld=outtbl.FindField("Release2")

wsfld=outtbl.FindField("Release3")

otherfld=outtbl.FindField("Release4")

wthdrfld=outtbl.FindField("Total")

theTbl.GetWin.Open

  for each recs in 1..damrecs

    rec=outtbl.AddRecord

    outtbl.SetValue(irrifld,rec,0)

    outtbl.SetValue(indfld,rec,0)

    outtbl.SetValue(wsfld,rec,0)

    outtbl.SetValue(otherfld,rec,0)

    outtbl.SetValue(wthdrfld,rec,0)

  end

outtbl.SetEditable(true)

dmdfldlst={ irrifld,indfld,wsfld,otherfld,wthdrfld}

  Sourcetbl=msgbox.choiceAsString(namelist,"Select a source table
containing"+nl+"reservoir release data", "Source Table")

```

```

if(sourcetbl=nil)then

exit

end

srctbl=theproject.finddoc(sourcetbl.asstring)

if(srctbl=nil)then

msgbox.info("Add source table "+sourcetbl.asstring+"to the project","")

exit

end

srcvtab=srctbl.getvtab

if(srctbl=nil)then

msgbox.info("Source table "+sourcetbl.asstring+"not found","")

exit

end

m=0

While(m<>99)

srcfields=srcvtab.getfields

Sourcefld=msgbox.choiceAsString(srcfields,"Select a source field containing
a"+nl+" component of total release","Source Field")

```

```

    if(sourcefld=nil)then

        exit

    end

    Targetfld=msgbox.choiceAsString(dmdfldlst,"Select a target field in
release.dbf","Targetfld")

    if(targetfld=nil)then

        msgbox.info("No target field selected","")

        exit

    end

    flowlist=list.make

    srcrcs=srcvtab.getnumrecords

    for each rec in 0..(srcrcs-1)

        myvalue=srcvtab.returnvalue(sourcefld,rec)

        flowlist.add(myvalue)

    end

    if(flowlist.isEmpty)then

        msgbox.info("No records found in source field","")

    exit

```



```

end

for each year in 0..(numyrs-1)

    drec=year*12

    lagmon=11

    if(year=(numyrs-1))then

        lagmon=mylag-1

    end

    'msgbox.info("Lagmon = "+lagmon.asstring,"")

    for each month in 0..lagmon

        drec=drec+1

        outtbl.SetValue(targetfld,drec-1,flowlist.get(month))

        thetbl.GetWin.Invalidate

    end

end

check=msgbox.YESNO("Do you want to input data "+nl+"for more release
fields?","check.ave",false)

if(check=false)then

    m=99

```

```

        end

    end

    for each srec in 1..damrecs

        irridmd=outtbl.ReturnValue(irrifld,srec-1)

        inddmd=outtbl.ReturnValue(indfld,srec-1)

        wsdmd=outtbl.ReturnValue(wsfld,srec-1)

        otherdmd=outtbl.ReturnValue(otherfld,srec-1)

        withdraw=irridmd + inddmd+ wsdmd + otherdmd

        outtbl.SetValue(wthdrfld,srec-1,withdraw)

    end

    thetbl.GetWin.Invalidate

    chkupdate=msgbox.YESNO("Do you want to update release in
    "+thedam.asstring+" now?", "Update Release?",true)

    damwthdr=theDam.FindField("Release")

    if(chkupdate=true)then

        av.ShowStopButton

        av.ShowMsg("Updating Release...")

        for each srec in 1..damrecs

```

```

        updwthdr=outtbl.ReturnValue(wthdrfld,srec-1)

        thedam.SetValue(damwthdr,srec-1,updwthdr)

    end

end

av.ClearStatus

end 'end of yes compute release

*****

Written 5/13/97

Kwabena Asante

CRWR, Univ. of Texas at Austin

```

Appendix 12: Script for Computing Simulation, Cmpsumry.ave

This program is used for updating values in the reservoir operation table

It creates summary.dbf for storing the reservoir summarized results of the reservoir simulation

The number of months in which spillage occurs is compared with the corresponding value for deficit

The total spillage and total deficit are computed for the entire simulation period

The percentage of inflow going to evaporation, leakage and withdrawal is also computed

The percentage change in reservoir storage is also computed

```
theproject=av.getproject
```

```
theTable=theProject.finddoc("dams.dbf")
```

```
theVtab=theTable.GetVtab
```

```
if(theVtab.IsEditable)then
```

```
    theVtab.seteditable(true)
```

```

end

thetbl=theProject.finddoc("summary.dbf")

if(thetbl<>nil)then

theproject.removedoc(thetbl)

end

thename="summary.dbf"

outtbl1=Vtab.MakeNew(theName.asfilename,dbase)

theTbl=Table.Make(outtbl1)

thetbl.setname("summary.dbf")

av.GetProject.AddDoc(theTbl)

Nfldlst=list.make

Nfldlst.Add(field.Make("Damid",#field_decimal,10,0))

Nfldlst.Add(field.Make("TotalMon",#field_decimal,10,0))

Nfldlst.Add(field.Make("DeficitMon",#field_decimal,10,0))

Nfldlst.Add(field.Make("SpillMon",#field_decimal,10,0))

Nfldlst.Add(field.Make("TotalInflow",#field_decimal,16,0))

Nfldlst.Add(field.Make("TotalDeficit",#field_decimal,16,0))

Nfldlst.Add(field.Make("TotalSpill",#field_decimal,16,0))

```

```

Nfldlst.Add(field.Make("Evap%",#field_decimal,10,5))

Nfldlst.Add(field.Make("Leak%",#field_decimal,10,5))

Nfldlst.Add(field.Make("Release%",#field_decimal,10,5))

Nfldlst.Add(field.Make("Withdraw%",#field_decimal,10,5))

Nfldlst.Add(field.Make("ChngStor%",#field_decimal,10,5))

outlist=Nfldlst.DeepClone

outtbl1.AddFields(outlist)

outtbl=thetbl.GetVtab

outtbl.SetEditable(true)

damidfld=outtbl.FindField("Damid")

totmon=outtbl.FindField("TotalMon")

defctmon=outtbl.FindField("DeficitMon")

spillmon=outtbl.FindField("SpillMon")

tinfld=outtbl.FindField("TotalInflow")

spfld=outtbl.FindField("TotalSpill")

tdefctfld=outtbl.FindField("TotalDeficit")

devapfld=outtbl.FindField("Evap%")

dleakfld=outtbl.FindField("Leak%")

```

```

drelsflid=outtbl.FindField("Release% ")

dwthdrfld=outtbl.FindField("Withdraw% ")

dcstorfld=outtbl.FindField("ChngStor% ")

thebitmap=outtbl.getselection

thebitmap.setall

outtbl.updateselection

if(thebitmap.isempty.not)then

outtbl.RemoveRecords(outtbl.getselection.clone)

end

damlst=list.make

for each rec in theVtab

    dmid=theVtab.FindField("damid")

    damIdNum=theVtab.returnvalue(dmid,rec)

    damlst.add(damIdNum)

end

if(damlst.isempty)then

msgbox.info("No Dams found in Dams.dbf", "")

exit

```

```

end

rectoadd=theVtab.getnumrecords

for each rec in 0..(rectoadd-1)

    rec=outtbl.AddRecord

end

damstoedit=0

for each damnum in damlst

    damstoedit=damstoedit+1

    mydam=theProject.finddoc("Dam"+damNum.asString+".dbf")

    if(myDam=nil)then

        msgbox.info("Dam"+DamNum.asstring+".dbf"+"Not Found","")

        exit

    end

    thedam=myDam.GetVtab

    if(theDam.IsEditable)then

        theDam.seteditable(true)

    end

    damrecs=theDam.GetNumRecords

```


dmspill=thedam.FindField("Spill")

dmdfct=thedam.FindField("Deficit")

dminflow=thedam.FindField("Inflow")

dmevap=thedam.FindField("Evaploss")

dmleak=thedam.FindField("Leakage")

dmwithd=thedam.FindField("Withdraw")

dmrels=thedam.FindField("Release")

dmstor=thedam.FindField("Storage")

spillcount=0

defctcount=0

spillall=0

defctall=0

inflowall=0

evapall=0

leakall=0

withdall=0

relsall=0

av.ShowStopButton

av.ShowMsg("Computing Reservoir Simulation Summary...")

for each rec in 1..damrecs

spillfld=thedam.ReturnValue(dmspill,rec-1)

defctfld=thedam.ReturnValue(dmdfct,rec-1)

inflowfld=thedam.ReturnValue(dminflow,rec-1)

evapfld=thedam.ReturnValue(dmevap,rec-1)

leakfld=thedam.ReturnValue(dmleak,rec-1)

withdfld=thedam.ReturnValue(dmwithd,rec-1)

relsfld=thedam.ReturnValue(dmrels,rec-1)

spillall= spillall+spillfld

defctall=defctall+defctfld

inflowall=inflowall+inflowfld

evapall= evapall+evapfld

leakall=leakall+leakfld

withdall=withdall+withdfld

relsall=relsall+relsfld

```

    if(spillfld>0)then

        spillcount=spillcount+1

    end

    if(defctfld<0)then

        defctcount=defctcount+1

    end

end

damstorage0=theVtab.FindField("Storage0")

damsBitMap=theVtab.GetSelection

damsBitMap.clearall

expr="([Damid] = "+damnum.asstring+" )"

theVtab.Query(expr,damsBitMap,#VTAB_SELTYPE_NEW)

theVtab.UpdateSelection

for each rec in damsbitmap

    inistorage=theVtab.ReturnValue(damstorage0,rec)

end

finalstorage=thedam.ReturnValue(dmstor,(damrecs-1))

changestor=(finalstorage-inistorage)

```

```

outtbl.SetEditable(true)

    outtbl.SetValue(damidfld,damstoedit-1,damnum)

    outtbl.SetValue(totmon,damstoedit-1,damrecs)

    outtbl.SetValue(spillmon,damstoedit-1,spillcount)

    outtbl.SetValue(defctmon,damstoedit-1,defctcount)

    outtbl.SetValue(tdefctfld,damstoedit-1,defctall)

    outtbl.SetValue(spfld,damstoedit-1,spillall)

    outtbl.SetValue(tinflld,damstoedit-1,inflowall)

    outtbl.SetValue(devapfld,damstoedit-1,(evapall*100/inflowall))

    outtbl.SetValue(dleakfld,damstoedit-1,(leakall*100/inflowall))

    outtbl.SetValue(dwthdrfld,damstoedit-1,(withdall*100/inflowall))

    outtbl.SetValue(drelsfld,damstoedit-1,(relsall*100/inflowall))

    outtbl.SetValue(dcstorfld,damstoedit-1,(changestor*100/inistorage))

end

outtbl.SetEditable(false)

thetbl.GetWin.Invalidate

theTbl.GetWin.Open

av.clearstatus

```

Appendix 13: Reservoir Simulation Script, Damrt.ave

'This program takes ToSub array
'from SFlowSim.prc (SurfaceRouting)
'and routes the dam object

Dname="Dams.shp"

theProject=av.getProject
theView=av.getactivedoc
DamTheme=theView.findTheme(Dname)

if(Damtheme=nil)then
 msgbox.info(Dname,"NotFound")
 exit
end

if(DamTheme.Is(Ftheme).not)then
 msgbox.error(Dname++"IsNotFtheme","")
 exit
end

DamFtab=DamTheme.GetFtab

initializes Field pointers

```
Dshape=DamFtab.FindField("shape")
DamId=DamFtab.FindField("Damid")
DamName=DamFtab.FindField("DamName")
DamCapt=DamFtab.FindField("Capacity")
DamArea=DamFtab.FindField("Area")
DamUpst=DamFtab.FindField("Upst")
DamDead=DamFtab.FindField("DeadSt")
DamEvt=DamFtab.FindField("Evt")
DamPdam=DamFtab.FindField("Pdam")
DamNdam=DamFtab.FindField("Ndam")
DamArea0=DamFtab.FindField("Area0")
DamSt0=DamFtab.FindField("Storage0")
DamGCode=DamFtab.FindField("Grid-Code")
DamPcnt=DamFtab.FindField("Pcntage")
Rlength=DamFtab.FindField("Length")
```

initializeDamDictionaryToCreatethe

RelationshipBetweenRecnumAndDamId

```
DamDict=Dictionary.make(DamFtab.GetNumRecords)
For each Drec in DamFtab
  DamDict.Add(DamFtab.ReturnValue(DamId,Drec),Drec.AsString.AsNumber)
end
```

'Getting Info From Main Program

DamidV=self.get(0)

IsHeadV=self.get(1)

Istep=Self.get(2)

LV=Self.get(3)

IniTmStep=Self.get(7)

EndTmStep=Self.get(8)

FFwlist=list.make

PFwlist=list.make

Ulist=list.make

for each i in 0..(EndTmStep-IniTmStep)

FFwlist.add(Self.get(4).get(i))

PFwlist.add(self.get(5).get(i))

end

'for each i in 0..Nstep

,

chk=msgbox.input("FW="+FFwlist.get(i).AsString+"p:"+PFwlist.get(i).AsString,

Ffwlist.Count.AsString,Pfwlist.count.Asstring)

' if(chk=nil)then

' break

' end

'end

'SetupRiverRelatedInformation

RlengthV=DamFtab.ReturnValue(Rlength,DamDict.Get(DamidV))

```

ErrK=0
OldPcntV=0
*****

'ThisLoopSimulatesAllTheReservoirsLocated
'InThisRiverSection
'IfNextDam=0,ExitTheLoop
*****

While(True)
ErrK=ErrK+1
if(ErrK>100)then
    msgbox.info("ERRK>100","EXIT->FromDamrt.ave")
    break
end

*****

'And Getting Dam's Related Info
'evtV in m/month
'TheUnit of Evaporation will be not be worked on
*****

DamRecNum=DamDict.Get(DamidV)
MaxAreaV=DamFtab.ReturnValue(Damarea,DamDict.Get(DamidV))
MaxStV=DamFtab.ReturnValue(DamCapt,DamDict.Get(DamidV))
EvtV=DamFtab.ReturnValue(DamEvt,DamDict.Get(DamidV))
PcntV=DamFtab.ReturnValue(DamPcnt,DamRecNum)
St0=DamFtab.ReturnValue(DamSt0,DamRecNum)
Sarea0=DamFtab.ReturnValue(DamArea0,DamRecNum)

```



```

DeadStV=DamFtab.ReturnValue(DamDead,DamRecNum)
CPcnt=(PCntV-OldPcntV)
LengV=CPcnt*RlengthV
Llag=LengV/(LV*ISTEP)
*****

`Chking to make sure that data are correctly
`passed from the main-program and read from table
*****

`chklist=list.make
`chklist.add("DmR="+DamRecNum.Asstring)
`chklist.add("MaxSt"++MaxStV.AsString)
`chklist.add("MaxA"++MaxAreaV.AsString)
`chklist.add("EvtV"++EvtV.AsString)
`chklist.add("PcntV"++PcntV.AsString)
`chklist.add("St0"++St0.AsString)
`chklist.add("Cpcnt"++Cpcnt.AsString)
`chklist.add("LengV"++LengV.AsString)
`chklist.add("Llag"++Llag.AsString)
`chk=msgbox.choiceasString(chklist,LV.ASString,OldPcntV.AsString)
Ulist.empty
Ulist.add(1-LLag)
Ulist.add(LLag)
UM=2
*****

`SetupDamTable->TimeSeries
`I have used individual tables instead of the generalized table
`I have also changed the mode of data entry in the dam locator program

```

```
ThisDam="Dam"+DamidV.AsString+".dbf"
DamTbl=TheProject.FindDoc(thisDam)
if(Damtbl=nil)then
  MsgBox.info(thisDam+"NotFound"+nl+"-
>Add"+ThisDam+"totheProject","EXIT5")
  exit
end

DamVt=DamTbl.GetVtab
MsgBox.info("This is Dam"++DamidV.AsString,"check.ave")
if(DamVt=nil)then
  MsgBox.info(thisDam+"NotFound"+nl+"-
>Add"+ThisDam+"totheProject","EXIT5")
  exit
end
If(DamVt.CanEdit)then
  DamVt.SetEditable(true)
end
DvtWithd=DamVt.FindField("WithDraw")
DvtNEvp=DamVt.FindField("Netevap")
DvtArea=DamVt.FindField("SurfArea")
DvtLoss=DamVt.FindField("EvapLoss")
DvtSpil=DamVt.FindField("Spill")
DvtSt=DamVt.FindField("Storage")
Dvtime=DamVt.Findfield("Time")
```

```

DvtHead=DamVt.Findfield("Head")
Dvtleak=DamVt.Findfield("Leakage")
Dvtrels=DamVt.Findfield("Release")
DvtCinflow=DamVt.Findfield("Inflow")
Dvtdeficit=DamVt.Findfield("Deficit")
Dvtinflws=DamVt.Findfield("Inflows")

```

```

*****

```

```

'SetupDictionaryForTime

```

```

*****

```

```

if(ErrK=1)then
  DmtmDict=Dictionary.make(DamVt.GetNumRecords)
else
  DmTmDict.Empty
end
for each rec in DamVt
  tmv=DamVt.ReturnValue(DvTime,rec)
  DmtmDict.add(tmv,rec.AsString.AsNumber)
end

```

```

'Inserted a routine here for determining the dam area and elevation from the
storage

```

```

CDam="Dmrg"+DamidV.AsString+".dbf"

```

```

DmTbl=TheProject.FindDoc(CDam)

```

```

if(DmTbl=nil)then

```

```

  MsgBox.info(CDam++"NotFound"++"->Using equation","equation.ave")

```

```

  DamAreaV=MaxAreaV*((St0/MaxStV)^(0.72))

```

```
DamFtab.SetEditable(true)
DamFtab.SetValue(DamArea0,DamRecNum,DamAreaV)
DamFtab.SetEditable(false)
```

```
else
```

```
DmVt=DmTbl.GetVtab
```

```
CDvol=DmVt.FindField("Storage")
CDarea=DmVt.FindField("Area")
CDhead=DmVt.FindField("Head")
CDindex=DmVt.FindField("Index")
Cdrec=DmVt.GetNumRecords
```

```
if(ErrK=1)then
  DmDict=Dictionary.make(DmVt.GetNumRecords)
else
  DmDict.Empty
end
```

```
for each rec in DmVt
  ind=DmVt.ReturnValue(CDindex,rec)
  DmDict.add(ind,rec.AsString.AsNumber)
end
```

```
for each rec in DmVt
  cdv=DmVt.ReturnValue(CDvol,rec)
```

```

cda=DmVt.ReturnValue(CDarea,rec)
cdh=DmVt.ReturnValue(CDhead,rec)
if(St0<=cdv)then
    pdv=DmVt.ReturnValue(CDvol,rec-1)
    pda=DmVt.ReturnValue(CDarea,rec-1)
    pdh=DmVt.ReturnValue(CDhead,rec-1)
    if(St0>pdv)then
        DamAreaV=pda+((cda-pda)*(St0-pdv)/(cdv-pdv))
        DamHeadV=pdh+((cdh-pdh)*(St0-pdv)/(cdv-pdv))
        DamFtab.SetEditable(true)
        DamFtab.SetValue(DamArea0,DamRecNum,DamAreaV)
        DamFtab.SetEditable(false)

    end
end
end

end

for each i in 0..(EndTmStep-IniTmStep)
    IFlowV=0.0

    if(i<(UM-1))then
        for each k in 0..i
            if(i=0)then
                IFlowV=IFlowV+FFwlist.get(0)
                'Sarea=Sarea0
            else

```

```

    IFlowV=IFlowV+((FFwlist.get(i-k))*(Ulist.get(k)))
    `Sarea=DamVt.ReturnValue(DvtArea,DmtmDict.get(i+IniTmStep-1))

    end
    end `for each k in 0..i
else
    for each k in 0..(UM-1)
        IFlowV=IFlowV+((FFwlist.get(i-k))*(Ulist.get(k)))
    end
end `if(i<(UM-1))

if(i=0)then
    Sarea=Sarea0
else
    Sarea=DamVt.ReturnValue(DvtArea,DmtmDict.get(i+IniTmStep-1))

end

PFlowV=(PFwList.get(i))*Cpcnt
IFlowV=IFlowV+PFlowV
InflowsV=DamVt.ReturnValue(Dvtinflws,DmtmDict.get(i+IniTmStep))
Qout=DamVt.ReturnValue(Dvtrels,DmtmDict.get(i+IniTmStep))
DwithV=DamVt.ReturnValue(DvtWithD,DmtmDict.get(i+IniTmStep))
Dleak=DamVt.ReturnValue(Dvtleak,DmtmDict.get(i+IniTmStep))
EvtV=DamVt.ReturnValue(DvtNevp,DmtmDict.get(i+IniTmStep))

EvplLoss=EvtV*Sarea

```

```

tout=(DwithV+Dleak+Evploss+Qout)
therelease=Qout

'StV=((IFlowV*Istep*30.4)-tout+ St0)
StV = (InflowsV-tout+St0)

numruns=0

for each numruns in 0..5

' if(numruns=9)then
    StMean=(StV+St0)/2

    if(DmTbl=nil)then
        MsgBox.info(CDam+"NotFound"+"->Using equation","equation.ave")
        DamAreaV=MaxAreaV*((StMean/MaxStV)^(0.72))
        EvpLoss=EvtV*DamAreaV
        tout=(DwithV+Dleak+Evploss+Qout)
        StV = (InflowsV-tout+St0)
    else
        DmVt=DmTbl.GetVtab

        CDvol=DmVt.FindField("Storage")
        CDarea=DmVt.FindField("Area")
        CDhead=DmVt.FindField("Head")
        CDindex=DmVt.FindField("Index")
        Cdrec=DmVt.GetNumRecords

```

```

for each rec in DmVt
  cdv=DmVt.ReturnValue(CDvol,rec)
  cda=DmVt.ReturnValue(CDarea,rec)
  cdh=DmVt.ReturnValue(CDhead,rec)
  if(StMean<=cdv)then
    if (rec=0)then
      pdv=0
      pda=0
      pdh=0
    else
      pdv=DmVt.ReturnValue(CDvol,rec-1)
      pda=DmVt.ReturnValue(CDarea,rec-1)
      pdh=DmVt.ReturnValue(CDhead,rec-1)
    end
    if(StMean>pdv)then
      DamAreaV=pda+((cda-pda)*(StMean-pdv)/(cdv-pdv))
      DamVt.SetValue(DvtArea,DmtmDict.get(i+IniTmStep),DamAreaV)
      EvpLoss=EvtV*DamAreaV
      tout=(DwithV+Dleak+Evploss+Qout)
      StV = (InflowsV-tout+St0)
    end
  end ' if(StMean<=cdv)
end 'if(DmTbl=nil)

'end 'if(numruns=1)

numruns=numruns+1

```



```

end ' for each numruns in 0..1

end ' for each i in 0..(EndTmStep-IniTmStep)

Qt=0      'Initialize spill
deficit=0  'Initialize deficit

'remember: IStep = 24*3600

if(StV>MaxStV)then
  Qt=(StV-MaxStV)
  therelease=(Qout+Qt)
  StV=MaxStV
end

if(StV<DeadStV)then
  deficit=-(DeadStV-StV)
  therelease=Qout+deficit

  StV=DeadStV

  if(therelease<0)then
    therelease=0
  end
end
end

```

'This is where I have put in the volume-area-elevation curve data

'This is the synthetic equation: $\text{DamAreaV} = \text{MaxAreaV} * ((\text{StV} / \text{MaxStV})^{0.72})$

if(DmTbl=nil)then

 MsgBox.info(CDam+"NotFound"+"->Using equation","equation.ave")

$\text{DamAreaV} = \text{MaxAreaV} * ((\text{Stv} / \text{MaxStV})^{0.72})$

else

DmVt=DmTbl.GetVtab

CDvol=DmVt.FindField("Storage")

CDarea=DmVt.FindField("Area")

CDhead=DmVt.FindField("Head")

CDindex=DmVt.FindField("Index")

Cdrec=DmVt.GetNumRecords

for each rec in DmVt

 cdv=DmVt.ReturnValue(CDvol,rec)

 cda=DmVt.ReturnValue(CDarea,rec)

 cdh=DmVt.ReturnValue(CDhead,rec)

 if(StV<=cdv)then

 if (rec=0)then

 pdv=0

 pda=0

 pdh=0

 else

 pdv=DmVt.ReturnValue(CDvol,rec-1)

```

    pda=DmVt.ReturnValue(CDarea,rec-1)
    pdh=DmVt.ReturnValue(CDhead,rec-1)
end
    if(StV>pdv)then
        DamAreaV=pda+((cda-pda)*(Stv-pdv)/(cdv-pdv))
        DamHeadV=pdh+((cdh-pdh)*(Stv-pdv)/(cdv-pdv))
        DamVt.SetValue(DvtHead,DmtmDict.get(i+IniTmStep),DamHeadV)
        DamVt.SetValue(DvtArea,DmtmDict.get(i+IniTmStep),DamAreaV)

    end
end
end

end

    DamVt.SetValue(Dvtspil,DmtmDict.get(i+IniTmStep),Qt)

'DamVt.SetValue(DvtCinflow,DmtmDict.get(i+IniTmStep),(IFlowV*Istep*30.4))
    DamVt.SetValue(DvtSt,DmtmDict.get(i+IniTmStep),StV)
    DamVt.SetValue(DvtLoss,DmtmDict.get(i+IniTmStep),EvplLoss)
'DamVt.SetValue(Dvtrels,DmtmDict.get(i+IniTmStep),therelease)
    DamVt.SetValue(Dvtdeficit,DmtmDict.get(i+IniTmStep),deficit)

    St0=StV
    end  'for each i in 0..NStep (EndTmStep-iniTmStep)
DamVt.SetEditable(False)
NextDam=DamFtab.ReturnValue(DamNdam,DamRecNum)
if(nextDam=0)then

```

```

LLag=(1-PcntV)*RLengthV/(LV*Istep)
FFwList.empty
For each i in 0..(EndTmStep-IniTmStep)
    FFwList.add((DamVt.ReturnValue(Dvtrels,DmtmDict.get(i+IniTmStep))/Istep))
end
break
else
    DamidV=NextDam
    OldPcntV=PcntV
    DamRecNum=DamDict.Get(DamidV)
    MaxAreaV=DamFtab.ReturnValue(Damarea,DamRecNum)
    MaxStV=DamFtab.ReturnValue(DamCapt,DamRecNum)
    EvtV=DamFtab.ReturnValue(DamEvt,DamRecNum)
    PcntV=DamFtab.ReturnValue(DamPcnt,DamRecNum)
    St0=DamFtab.ReturnValue(DamSt0,DamRecNum)
    CPcnt=(PCntV-OldPcntV)
    LengV=CPcnt*RlengthV
    Llag=LengV/LV

    CDam="Dmrg"+DamidV.AsString+".dbf"
    DmTbl=TheProject.FindDoc(CDam)
    if(dmtbl<>nil)then
        DmVt=DmTbl.GetVtab
    end

    ThisDam="Dam"+DamidV.AsString+".dbf"
    DamTbl=TheProject.FindDoc(thisDam)
    if(damtbl=nil)then

```

```

    MsgBox.info(thisDam+"NotFound"+nl+"-
>Add"+ThisDam+"totheProject","EXIT5")
    exit
end

```

```

DamVt=DamTbl.GetVtab
if(damVt=nil)then
    MsgBox.info(thisDam+"NotFound"+nl+"-
>Add"+ThisDam+"totheProject","EXIT5")
    exit
end

```

```

end
end While(true)

```

```

DamFtab.SetEditable(False)
return({FFwList,LLag})

```

Bibliography

- 1) Connor, S. Et al, (1993) Integrated GIS and Hydrologic Modeling for Countywide Drainage Study, *Journal of Water Resource Planning and Management*, 119(2), pp112-128
- 2) Chow, V.T., Maidment, D.R., Mays, L.W. (1987), Applied Hydrology, McGraw-Hill Book Company, NY
- 3) Environmental Systems Research Institute (1992), Understanding GIS: The Arc/Info Method, ESRI, Redlands, CA
- 4) Hsu, N-S, Kuo, J-T, Chu, W-S, Lin Y-J (1995), Proposed Daily Streamflow Forecasting Model for Reservoir Operation, *Journal of Water Resource Planning and Management*, 121(2), pp132-
- 5) Hufschmidt M.M. Et al (1966), Simulation Techniques for Design of Water Resource Systems, Harvard University Press, Cambridge, Massachusetts
- 6) Kilgore, R.T., Krolak, J.S., Walsh, M. (1994), Integration of Water resource Models and GIS for Water Resource Management, Proceedings of the 21st Annual Conference on Water Policy and Management: Solving the Problems, ASCE, NY NY

- 7) Kim, H-B. (1996), Capacity Expansion Modeling of Water Supply in a Planning Support System for Urban Growth Management, University of Illinois at Urbana-Champaign
- 8) Kouwen, N. Et al (1993), Grouped Response Unit for Distributed Hydrologic Modeling, *Journal of Water Resource Planning and Management*, 119(3), pp289-305
- 9) Loucks,D.P., Stedinger, J.R., Haith, D.A. (1981), Water Resource Planning and Analysis, Pretice-Hall, Inc., Englewood Cliffs, NJ
- 10) Lullwitz, T., Helbig, A. (1994), Grid Related Estimates of Streamflow within the Weser River Basin, Germany, Modeling and Management of Sustainable Basin-Scale Water Resource Systems, IAHS, Wallingford, UK
- 11) Maidment, D.R. (Edited by) (1992), Handbook of Hydrology, McGraw-Hill, Inc., NY
- 12) Mattingly, S.B. (1994), Using Geographical Information Systems to Manage a Multipurpose Water Project, Proceedings of the 21st Annual Conference on Water Policy and Management: Solving the Problems, ASCE, NY NY
- 13) Reed, S., Maidment, D.R., Patoux, J. (1997), Spatial Water Balance of Texas, Center for Research in Water Resources, the University of Texas at Austin
“<http://www.ce.utexas.edu/prof/maidment/gishydro/docs/wbtexas/wbtexas.htm>”

- 14) Ross, M., Tara, P. (1993) Integrated Hydrologic Modeling with Geographic Information Systems, *Journal of Water Resource Planning and Management*, 119(2), pp129-140
- 15) Sharma, N.P. Etal (1996), African Water Resources: Challenges and Opportunities for Sustainable Development, World Bank Technical Paper No. 331
- 16) Singh, K.P. (1995), Prediction of River Discharge at Ungaged Sites with Analysis of Uncertainty, *Journal of Water Resource Planning and Management*, 121(3), pp284
- 17) Takeuchi, K., Sivaarthitkul, V. (1995), Assessment of effectiveness of the use of inflow Forecasts to Reservoir Management, Proceedings of a Boulder Symposium on Modeling and Management of Sustainable Basin-Scale Water Resource Systems, IAHS 231, pp299-304
- 18) Teegarden, T., Pochop L., Kerr G., Delaney, R. (1994), State Wide Consumptive Use Estimates, Proceedings of the 21st Annual Conference on Water Policy and Management: Solving the Problems, ASCE, NY
- 19) United Nations Conference on Environment and Development (1992), Report of the United Nations Conference on Environment and Development, A/CONF.151/26 Vol.II, Chapter 18, Rio de Janeiro, Brazil
- “gopher://gopher.un.org:70/11/conf/unced/English”

- 20) Walsh, M.R. (1993), Toward Spatial Decision Support Systems in Water Resources, *Journal of Water Resource Planning and Management*, 119(2), pp158-169
- 21) Wurbs, R.A. (1993), Reservoir-System Simulation and Optimization Models, *Journal of Water Resource Planning and Management*, 119(4), pp455-472
- 22) Ye, Z., Maidment, D.R., McKinney, D.C. (1996), Map-based Surface and Subsurface Flow Simulation Models: An Object-Oriented and GIS Approach, Center for Research in Water Resources, the University of Texas at Austin
- 23) Yeh, W. W-G, (1985), Reservoir Management and Operations Models: A State-of-the-art Review, *Water Resource Research*, 21(12), pp1797-1818

VITA

Kwabena Oduro Asante was born in Kintampo, Ghana on November 23, 1971, the son of Beatrice Oduro Asante and Kofi Oduro Asante. After completing his work at Starehe Boys Centre and School, Nairobi, Kenya, in 1989, he studied computer programming at the Kestrel College, Nairobi. He entered the University of Nairobi in August of 1990 and received the degree of Bachelor of Science in Civil Engineering in December, 1994. He served as an Engineering Trainee at the Water Resources Research Institute in Accra, Ghana, before entering the Graduate School at The University of Texas in August, 1995.

Permanent Address: 24 Route de Sauverny
 1290 Versoix
 Geneva, Switzerland

This thesis was typed by the author.